“十三五”国家重点出版物出版规划项目
高等教育网络空间安全规划教材

现代密码学

任伟　许瑞　宋军　编著

机 械 工 业 出 版 社

本书内容包括密码学概述、古典密码体制、信息理论安全、序列密码、分组密码、Hash 函数和消息鉴别、公钥加密（基础）、公钥加密（扩展）、数字签名、实体认证与身份识别、密码管理。本书的特色是注重介绍密码学方案设计的基本原理和内在规律，展现知识体系之间的内在逻辑性，大量采用类比和比较的方式探究方法论，力图使读者“知其所以然”。在给出方案的同时，解释方案的设计机理和思路，并专门设立“思考”环节，有意识地培养读者的创造性思维能力。

本书读者对象包括高等院校网络空间安全、信息安全、密码学、应用数学、计算机科学、信息与计算科学等专业本科高年级学生。对信息安全领域的研究人员也具有启发作用和参考价值。

本书配有授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取。微信：15910938545。电话：010-88379739。

图书在版编目（CIP）数据

现代密码学／任伟，许瑞，宋军编著．—北京：机械工业出版社，2020.8
“十三五”国家重点出版物出版规划项目　高等教育网络空间安全规划教材
ISBN 978-7-111-64867-3

Ⅰ．①现…　Ⅱ．①任…　②许…　③宋…　Ⅲ．①密码-理论-高等学校-教材　Ⅳ．①TN918.1

中国版本图书馆 CIP 数据核字（2020）第 032970 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
策划编辑：郝建伟　　责任编辑：郝建伟　陈崇昱　时　静
责任校对：张艳霞　　责任印制：常天培
北京捷迅佳彩印刷有限公司印刷

2020 年 9 月第 1 版・第 1 次印刷
184mm×260mm・15.25 印张・376 千字
0001-1500 册
标准书号：ISBN 978-7-111-64867-3
定价：55.00 元

电话服务
客服电话：010-88361066
　　010-88379833
　　010-68326294
封底无防伪标均为盗版

网络服务
机　工　官　网：www.cmpbook.com
机　工　官　博：weibo.com/cmp1952
金　　书　　网：www.golden-book.com
机工教育服务网：www.cmpedu.com

高等教育网络空间安全规划教材
编委会成员名单

前　言

目前市场上的密码学书籍依然有值得改进的地方，主要表现在虽然对密码方案过程的讲解较为细致，但是对方案的设计原理、设计原则等内在机制的讲解不够充分，导致读者学习了各种方案后，大多局限于对方案的代码实现和使用，没有深刻理解为什么要这样设计方案，特别是不知道密码学方案是如何想出来的，不知道其来龙去脉，没有理解方案设计中的内在逻辑性，因此，可认为没有“真正学懂”。同时，目前的相关书籍在思维的启发性、学生创造性能力培养方面仍然有待完善。本书力图在这些方面做出新的尝试，有意识地引导学生思考，培养他们的逻辑推理能力、发散性思维能力、知识的归纳能力，以及灵活运用所学知识的能力。

每章内容的组织是从整体全貌到局部细节，从一般模型到具体方案（先讲模型、分类，再介绍具体构造方案）。介绍具体内容时遵照学习规律，从易到难，从简单方案到改进方案，还原历史发展的原貌和变迁，突出来龙去脉（如在介绍公钥方案的提出时，首先介绍 Merkle 谜题，Pohlig-Hellman 对称密钥分组加密，Merkle-Hellman 背包公钥密码方案，然后才介绍 RSA 方案）。讲解内容时深入浅出，简洁直观，使用浅显的语言来表述深奥的内在规律。在介绍完多个具体方案后，再归纳一般规律，从具体方案中提炼出一般规律和原理（如从 ElGamal 签名、DSA 等到一般 ElGamal 签名，从基于身份识别协议到知识签名）。大部分方案都会给出实例进行直观的讲解。本书还大量采用类比法、比较法、归纳法、图示法，试图使读者对所学内容能够反复巩固、前后联系。写作本书时还特别遵循了以下思路：

1）注重启发性。目前大部分教材以罗列密码学方案为主，缺乏对设计原理的分析，以及对设计动机和逻辑性的解释。本书试图改变这一局面。

2）注重知识点的逻辑联系和类比。指出章节间和章节中前后各个分离的知识点间的联系和类比关系，明确给出各知识点间的关联，便于读者体会密码算法或协议设计的奥妙。

3）注重原理的总结和推广。在介绍完具体构造方案后，给出一般性构造方案，或者加以讨论和总结，给出提问思考，有利于知识的理解，起到举一反三的作用。

4）兼具广度和深度。基本原理和基本概念的讲解力求透彻、有深度。通过扩展阅读提高广度，便于读者回顾经典论文或者了解最新的国际国内发展动态。

全书共分 11 章：第 1 章为密码学概论；第 2 章介绍古典密码体制；第 3 章介绍信息理论安全；第 4 章介绍序列密码；第 5 章介绍分组密码；第 6 章介绍 Hash 函数和消息鉴别；第 7 章介绍公钥加密（基础）；第 8 章介绍公钥加密（扩展）；第 9 章介绍数字签名；第 10 章介绍实体认证与身份识别；第 11 章介绍密钥管理。

全书精心安排了示例。为帮助读者进一步对书中的内容进行拓展研究，本书还有针对性地提供了扩展阅读建议，用于开展课外学习和论文研读讨论。每章小结归纳了本章知识点，并指出重点和难点，便于复习。打 * 号的章节可选学。

本书第 3 章由许瑞编写，11.1 节和 11.4 节由宋军编写，其余章节由任伟编写，全书由任伟负责统稿和定稿。

本书的出版得到了省级教学研究项目（2015146）和本科教学质量工程项目（2016039）的支持，在此表示感谢。同时感谢学生肖睿阳的辅助性工作。由于编者水平有限，在此衷心希望读者批评指正，可将意见和建议反馈至 E-mail：weirencs@ cug.edu.cn。

编　者

目　录

第 1 章　密码学概论

本章要点：

- 密码学解决的主要目标问题。
- 从密码学的发展简史体会密码学方案设计的演变。
- 对加密体制的攻击和安全性。

1.1　密码学的目标与知识构成

随着信息社会的发展，信息安全成为一个需要解决的关键问题。针对信息安全的攻击，主要包括主动攻击和被动攻击。

被动攻击主要是信息的截取（Interception），指未经授权窃听传输的信息，企图分析出消息内容或者是通信模式。

主动攻击主要包括：

（1）中断（Interruption），阻止通信设施的正常工作，破坏可用性。

（2）篡改（Modification），更改数据流。

（3）伪造（Fabrication），将一个非法实体伪装成一个合法的实体。

（4）重放（Replay），将一个数据单元截取后进行重传。

思考 1.1：能否给出上述攻击的具体表现形式？ □

信息安全的目标是确保信息的安全性。安全目标通常包括：

（1）机密性（Confidentiality）。指保证信息不泄露给非授权的用户或者实体，确保保存的信息和被传输的信息仅能被授权的各方得到，而非授权用户即使得到信息也无法知晓信息的内容。通常通过访问控制机制阻止非授权用户的访问，通过加密机制阻止非授权用户知晓信息的内容。

（2）完整性（Integrity）。指消息未经授权不能进行篡改，要保证消息的一致性，即消息在生成、传输、存储和使用过程中不应发生人为（或无意）的非授权篡改（插入、修改、删除、重排序等）。一般通过访问控制阻止篡改行为，同时通过消息摘要算法来检测信息是否被篡改。

（3）认证性（Authentication）。指确保一个消息的来源或者消息本身被正确地标识，同时确保该标识没有被伪造，认证分为消息鉴别和实体认证。消息鉴别是指接收方保证消息确实来自于所声称的源；实体认证指能确保被认证实体是所声称的实体。

（4）不可否认性（Non-repudiation）。指能保证用户无法事后否认曾经对信息进行的生成、签发、接收等行为。当发送一个消息时，接收方能证实该消息确实是由既定的发送方发来的，称为源不可否认性；同样，当接收方收到一个消息时，发送方能够证实该消息确实已经送到了指定的接收方，称为宿不可否认性。一般通过数字签名来提供不可

否认服务。

（5）可用性（Availability）。指保障信息资源随时可提供服务的能力。即授权用户根据需要可以随时访问所需信息，保证合法用户对信息资源的使用不被非法拒绝。典型的对可用性的攻击是拒绝服务攻击。

思考 1.2：你能给出上述目标的具体例子吗？ □

除了以上一些主要目标外，还有隐私性（Privacy）、匿名性（Anonymity）等。

例 1.1 下面给出隐私性的几个例子。

移动互联网中的位置隐私（Location Privacy）：基于位置的服务（Location-based Service）通常需要用户的位置，而用户的位置通常又会泄露用户的隐私，因此，位置隐私需要保护。

大数据公开共享可能会导致用户数据的隐私泄露，因此大数据在公开的时候需要对数据进行处理，目前比较流行的做法是差分隐私方法。

比特币账本中的交易隐私：由于比特币账本是公开的，因此每一个人都可以查找账本和分析账本，于是有些交易可能会定位到某一个人，这个人的所有交易都能够被查找到，他的交易隐私也会因此被泄露。

思考 1.3：攻击和信息安全的目标是什么关系？

在实际中，我们通常先发现一种攻击，然后给出对该种攻击的防御措施。攻击可能针对多个信息安全的目标。每个信息安全的目标可能面临多种具体的攻击方式。将攻击方式的共同目标提炼和抽象出来，可以确定某些信息安全的目标。 □

为达到上述目标，信息安全采用了信息论、计算机科学和密码学等方面的知识，形成了一门综合学科，其主要任务是研究计算机系统和通信网络中信息的保护方法，以及实现系统和网络中信息的机密性、完整性、认证性、不可否认性、可用性等目标，其中密码学是实现信息安全目标的核心技术。

密码学（Cryptology）是研究实现信息安全各目标的相关的数学、方法和技术。密码学不是提供信息安全的唯一方式。其研究的目标是信息安全目标的一个子集，主要包括：机密性、完整性、认证性、不可否认性。（注意，这里没有包括可用性。）为实现上述目标，密码学集数学、计算机科学、电子与通信等诸多学科的方法于一体，是一门交叉学科。从大的方面可分为密码编码学和密码分析学两类，对应于密码方案的设计学科和密码方案的分析学科。

密码学在设计方案的时候，首先需要考虑方案所能达到的安全性。通常，衡量密码体制安全性的基本准则有以下几种：

（1）计算安全（Computational Security）：如果攻破一个密码体制所需要的计算能力和计算时间是现实条件所不具备的，就认为相应的密码体制满足计算安全。

（2）无条件安全（Unconditional Security）：如果假设攻击者在无限计算能力和计算时间的前提下，也无法攻破该体制，则认为相应的密码体制满足无条件安全。

（3）可证明安全（Provable Security）：如果攻破一个密码体制意味着可以解决某一个经过深入研究的数学难题，就认为相应的密码体制满足可证明安全。

思考 1.4：上述 3 个安全性基本准则之间的比较？

计算安全与计算能力和计算时间的假设有关，通常计算安全的安全程度是计算能力和计算时间的函数。例如，即使是当前世界 500 强的计算机也需要计算 100 年。

无条件安全不需要假设计算能力和计算时间。简言之，不定方程解在自变量定义范围内的取值是等可能的。多元线性方程组的个数（即线性方程组的秩）少于自变量的个数时，则线性方程组解在自变量定义范围内的取值是等可能的。即使对非经典的计算机，如量子计算机、DNA 计算机而言，也是安全的（即无法确定自变量的取值）。

可证明安全强调安全的规约，即安全性的缺失将导致某个数学难题的解决，换句话说，因为数学难题被广泛认为是不能解决的，因此安全性是可以保证的。 □

通常现代密码学强调达到可证明安全，这通常是计算安全的。即安全具有一定的等级，这种等级通常通过攻破方案所需要的工作量来衡量。一种衡量安全等级的常见参数就是密钥的长度。

除了安全性外，设计密码方案时还需要考虑如下因素：

（1）功能性。方案能够满足安全需求即可，要避免过于满足安全需求且性能代价过高的方案。也可以理解为“杀鸡不要用牛刀”。

（2）性能。方案的计算、存储、传输等各方面的效率。例如，嵌入式系统、手机系统、智能卡系统等在计算和存储能力上有限，设计安全方案时需要考虑到。

（3）容易实现性。在实际中实施方案的难易程度。包括在软件和硬件环境中实现密码要素的复杂度。

上述方面往往在实际应用中需要权衡，如在一个计算能力有限的环境中，为了使系统在整体上具有良好的性能，可能不得不割舍高级别的安全性。

围绕着密码学要达到的目标，可以将密码学的实现方案分类成各种工具。图 1.1 给出了密码学内容的构成，图 1.2 围绕着安全目标给出了各内容间的联系。

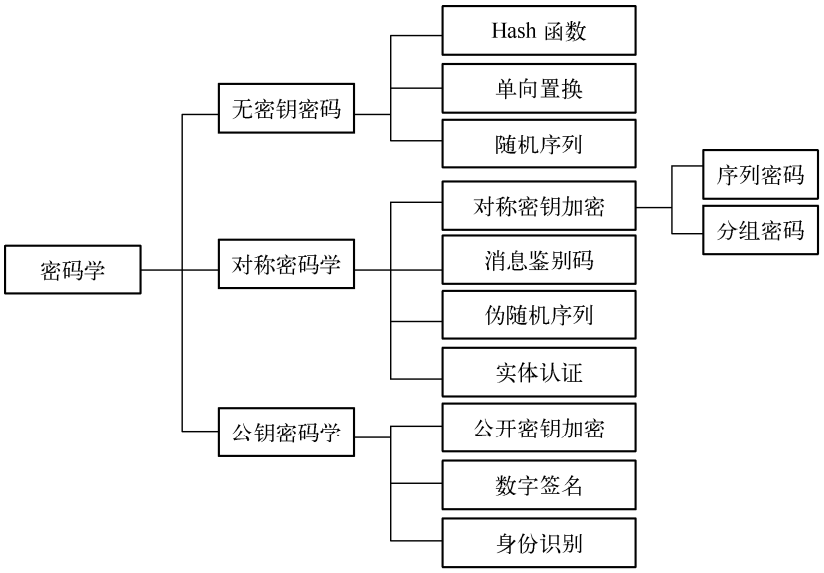


图 1.1　密码学的内容构成

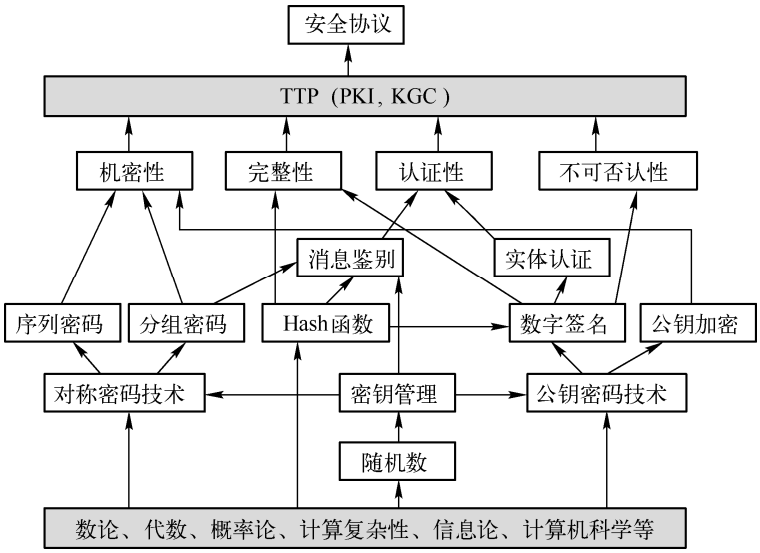


图 1.2　密码学研究内容间的关系

1.2　密码学的发展简史

从整体上来说，密码学经过了古典密码时期（人工密码）、近代密码时期（机械密码）、现代密码时期（电子计算机）这三个发展历程。下面按年代顺序列出密码学发展中的重要事件。

1.2.1　古典密码时期

（1）公元前 1900 年左右，一位佚名的埃及书吏在碑文中使用了非标准的象形文字，这或许是目前已知最早的密码术实例。

（2）公元前 400 多年，古希腊斯巴达军队中使用的 Scytale 密码，是一种置换密码。

（3）公元前 1 世纪，古罗马帝国皇帝凯撒（Caesar）曾经使用有序的单表代换密码，即 Caesar 密码，是单表代换密码的代表。

（4）我国宋代曾公亮、丁度等编撰的《武经总要·字验》中记载，北宋前期，在作战中曾用一首五言律诗的 40 个汉字，分别代表 40 种情况或要求，这种方式已具备了密码本的特点。

（5）欧洲的密码学起源于中世纪。约在 1379 年，欧洲第一本关于密码学的手册由生活在意大利北部城市帕尔玛的 Gabriela de Lavinde 写成，它由几个加密算法组成，且为罗马教皇克莱门特七世服务。

（6）阿拉伯人是第一个清晰地理解密码学原理的人，他们设计并使用代换和换位加密，并且发现了密码分析中的字母频率分布关系。大约在 1412 年，波斯人 al-Qalqashandi 所编的百科全书的第 14 卷中记载了破译简单代换密码的方法。这是密码分析法最早的著作之一。

（7）大约在 1467 年左右，意大利佛罗伦萨的建筑师 Alberti 发明了多字母表替代密码，他设计了一个密码盘，该盘有一个大一些的外轮和一个小一些的内轮，并各自以明文字符和密文字符作为索引。字母的排列确定了一个简单替代，并且可在加密一些字符之后通过转动

盘来修改替代方式。

（8）1508 年，密码学的第一本印刷书籍 Polygraphic 由德国的僧侣 Trithemius 写成，并在 1518 年出版，其中包含了第一个基于 24 个字符的方形表，该表列出了明文字母表字符在一个固定次序下的所有移位替代。

（9）17 世纪，英国著名的哲学家弗朗西斯·培根在他所著的《学问的发展》一书中最早给密码下了定义，他说："所谓密码应具备三个条件，即易于翻译、第三者无法理解、在一定场合下不易引人注意。"

（10）1854 年，Playfair 密码（Playfair Cipher）由 C. Wheatstone 提出，此后由他的朋友 L. Playfair 将该密码公布，所以就称为 Playfair 密码。

（11）1858 年，维吉尼亚密码（Vigenere Cipher）由法国密码学家 B. D. Vigenere 提出。

（12）1860 年，密码系统在外交通信中已得到普遍使用。如在美国南北战争中，联邦军广泛地使用换位加密，主要是使用 Vigenere 密码。

（13）1863 年，Kasiki 测试法于由普鲁士军官 F. Kasiski 提出，用于分析多表代换的周期。

（14）1871 年，上海大北水线电报公司选用 6899 个汉字，代以 4 码数字，成为中国最初的商用明码本，同时也设计了由明码本改编成密码本并进行混淆的方法。

（15）1883 年，A. Kerckhoffs 在《军事密码学》一书中提出了密码系统的安全性的一个基本假设，称为 Kerckhoffs 假设（原则），即密码分析者知道所使用的密码算法。

（16）1917 年，Vernam 密码由美国 AT&T 公司的 G. Vernam 提出，它是为电报通信设计的非常简单方便的密码，奠定了序列密码的基础。

（17）1918 年，W. F. Friedman 的论文《重合指数及其在密码学中的应用》（The Index of Coincidence and Its Applications in Cryptography），给出了多表代换密码的破译方法，它是 1949 年之前最重要的密码文献。

（18）1929 年，Hill 密码（Hill Cipher）由数学家 L. Hill 提出。

古典密码时期的密码技术仅是一门文字变换艺术，其研究和应用远没有形成一门科学，最多只能称其为密码术。

1.2.2 近代密码时期

（1）20 世纪 20 年代，随着机械和机电技术的成熟，以及电报和无线电需求的出现，引起了密码设备方面的一场革命——发明了转轮密码机（Rotor），转轮机的出现是密码学发展的重要标志之一，从此出现了商业密码机的公司和市场。

（2）从 1921 年开始的接下来的十多年里，美国加州奥克兰的一个名叫 Edward Hebern 的工程师构造了一系列改进的转轮机，并投入美国海军的试用评估，他申请了第一台转轮机的专利，这种装置在差不多 50 年内一直被指定为美军的主要密码设备，奠定了第二次世界大战中美国在密码学方面的超级地位。

（3）在美国的 Herbern 发明转轮密码机的同时，欧洲的工程师们如荷兰的 Hugo Koch、德国的 Arthur Scherbius 都独立地提出了转轮机的概念。德国的 Arthur Scherbius 于 1919 年设计出了历史上著名的密码机——ENIGMA 机（意思是"谜"）。1930 年，日本的第一台转轮密码机（美国分析家称之为 RED）开始为外交部门服务。1939 年，日本人引入了一台新的

加密机（美国分析家称之为 PURPLE），其中的转轮机用电话步进交换机取代。

（4）第二次世界大战是人工加密时代转变为机械加密时代的转折点。转轮密码机的大量使用极大地提高了加解密的速度，同时抗攻击性能也有很大的提高，是密码学发展史上的一座里程碑。同时，密码分析最伟大的成功发生在第二次世界大战期间，波兰人和英国人破译了 ENIGMA 密码，美国人攻破了日本的 RED、ORANGE 和 PURPLE 密码，对盟军在第二次世界大战中获胜起到了重要作用。

近代密码时期可以看作是科学密码学的前夜，这个阶段的密码技术可以说是一种艺术，也是一种技巧和经验的综合体，但还不是一种科学，密码专家常常是凭直觉和信念来进行密码设计和分析，而不是推理和证明。因此，也有学者将古典、近代密码时期划分为一个阶段。

1.2.3 现代密码时期

（1）1949 年，Shannon（香农）在 *Bell Systems Technical Journal* 上发表了《保密系统的通信理论》（*Communication Theory of Secrecy Systems*）一文，用概率和统计等科学工具研究加密系统，为密码学奠定了坚实的理论基础，从此密码学从艺术变为科学。

（2）1967 年，Kahn 出版了《破译者》（*The Codebreakers*）一书，对密码学的历史进行了相当完整的记述，使成千上万原本不知道密码学的人了解了密码学。从此，密码学研究引起了民间的兴趣。Kahn 认为是阿拉伯人创造了“加密法（cipher）”一词。

（3）1973 年，美国国家标准局（NBS，现在是美国国家标准技术研究所 NIST）在全世界范围征求国际密码标准方案（DES）。4 年后，发布正式的标准 DES。该方案的公布极大地促进了密码学在民间的研究。

（4）1976 年，W. Diffie 和 M. Hellman 在《密码学的新方向》一文中提出公钥密码体制。这是密码学发展史上最伟大的一次革命，也是现代密码学诞生的标志。

（5）1978 年，Merkle 和 Hellman 提出了第一个公钥密码系统——背包（knapsack）公钥密码系统，安全性基于背包问题（一种 NP 完全问题）。

（6）1978 年，美国麻省理工学院（MIT）的 Rivest、Shamir 和 Adleman 提出 RSA 加密机制，这是第一个实用的公钥方案，开创了密码学的新纪元。

（7）1979 年，MIT 的 M. O. Rabin 提出第一个可证明安全的公钥密码体制。

（8）1979 年，L. Lamport 提出基于任意单向函数的一次签名方案。

（9）1984 年，S. Goldwasser 与 S. Micali 提出了概率公钥密码系统的概念，并提出 Goldwasser-Micali 概率公钥密码系统。

（10）1984 年，IBM 公司的 Benett 和 Montreal 大学的 Brassard 提出第一个量子密码学方案，称为 BB84 协议。它是以量子力学基本理论为基础的量子信息理论领域的第一个应用，并提出了一个量子密钥交换的安全协议，由此迎来了量子密码学的新时期。

（11）1985 年，ElGamal 密码体制由 T. ElGamal 提出，该密码体制基于的困难问题是群中的离散对数问题。

（12）1985 年，T. ElGamal 提出一个基于离散对数问题的数字签名体制，称为 ElGamal 数字签名体制。

（13）1985 年，N. Koblitz 和 V. Miller 提出了椭圆曲线密码系统（Elliptic Curve Cryptog-

raphy，ECC），实现了公钥密码体制在效率上的重大突破。

（14）1987 年，R. Rivest 提出面向软件实现的序列密码 RC4，RC4 是目前公开范围内应用最广泛的序列密码。

（15）1988 年，Matsumoto 和 Imai 提出多变量公钥密码体制，这是第一个使用“小域——大域”的思想来构造域的方法，也是多变量公钥密码体制发展史上的里程碑，还是第一个实用的多变量公钥密码体制。

（16）1989 年，Robert A. J. Matthews 首次将混沌理论用于密码学研究，并提出一种基于变形 Logistic 映射的混沌序列密码方案。从此，混沌密码学作为密码学的一个分支引起了广泛的关注。

（17）1989 年，世界上第一台量子密钥分配原型样机研制成功，它的工作距离仅为 32cm，然而，它却标志着量子密码开始初步走向实用。

（18）1994 年，Peter Shor 发现了一种在量子计算机上多项式时间内运行的大整数因子分解算法，这意味着一旦人们能研制出量子计算机，则 RSA 密码体制将不再安全。

（19）1994 年，Adleman 利用 DNA 计算机解决了一个有向 Hamilton 路径问题，标志着信息时代进入了一个新的阶段。

（20）1996 年，Bellare 等人基于嵌套 MAC 提出 HMAC，并证明了其安全性。

（21）1996 年，在 Crypto 会议上，布朗大学的 Hoffstein、Pipher、Silverman 三位数学家提出了 NTRU（Number Theory Research Unit）公开密钥算法，它是一种基于格的快速公开密钥体制。

（22）1997 年，NIST 发起公开征集高级加密标准（Advanced Encryption Standard，AES）算法的活动。

（23）2000 年 10 月，美国政府在多次评审后宣布，比利时人发明的 Rijndael 算法为最终的 AES 算法。

（24）2001 年 1 月，欧洲委员会在信息社会技术（Information Society Technology，IST）规划中投入巨资，支持一项称为 NESSIE（New European Schemes for Signature，Integrity，and Encryption）的工程，希望通过公开征集和进行公开、透明的测试评估，推出一套安全性高、软硬件实现性能好、能适应不同应用环境的密码算法。该工程于 2003 年 2 月完成，极大地推动了密码学的研究。

（25）2004 年 2 月，欧洲委员会的 IST 基金支持了一个为期 4 年的项目，ECRYPT（European Network of Excellence for Cryptology），目标是促进欧洲信息安全研究人员在密码学和数字水印研究上的交流。2008 年该项目完成评审。

（26）2006 年，国家密码管理局公布了《无线局域网产品使用的 SMS4 密码算法》，该算法是我国自有知识产权的国际无线网络安全标准 WAPI 的一部分。这是我国第一次公布自己的商用密码算法。

（27）密码货币比特币（Bitcoin）的概念最初由中本聪在 2008 年 11 月 1 日提出，并于 2009 年 1 月 3 日正式诞生，从此出现了一门新的分支——密码经济学。

（28）2010 年 2 月，Kleinjung 等在 IACR 的 ePrint 预印版论文服务器上（2010/006）发表了论文 *Factorization of a 768-bit RSA modulus*，这是迄今分解的最大 RSA 模。

（29）2012 年 3 月 21 日，国家密码管理局公布了 6 项密码行业标准，包括：GM/T

0001—2012《祖冲之序列密码算法》、GM/T 0002—2012《SM4 分组密码算法》（原 SMS4 分组密码算法）、GM/T 0003—2012《SM2 椭圆曲线公钥密码算法》、GM/T 0004—2012《SM3 密码杂凑算法》、GM/T 0005—2012《随机性检测规范》、GM/T 0006—2012《密码应用标识规范》。为了保障商用密码的安全性，国家商用密码管理办公室还制定了一系列密码标准，包括 SM1（SCB2）、SM2、SM3、SM4、SM7、SM9、祖冲之密码算法（ZUC）等。其中 SM1、SM4、SM7、祖冲之密码算法（ZUC）是对称算法；SM2、SM9 是非对称算法；SM3 是哈希算法。

（30）2018 年，我国 SM2/3/9 密码算法正式成为 ISO/IEC 国际标准。同年 10 月，含有我国 SM3 杂凑密码算法的 ISO/IEC10118-3:2018《信息安全技术杂凑函数第 3 部分：专用杂凑函数》（第 4 版）由国际标准化组织（ISO）发布，SM3 算法正式成为国际标准。同年 11 月，作为补篇，2017 年就已被纳入国际标准的 SM2/SM9 数字签名算法，也以正文形式随 ISO/IEC14888-3：2018《信息安全技术带附录的数字签名第 3 部分：基于离散对数的机制》最新一版发布。

（31）2019 年 2 月 24 日，NIST 公布了 26 种算法进入后量子密码半决赛。17 个第二轮候选公钥加密和密钥建立算法是：BIKE、Classic McEliece、CRYSTALS-KYBER、FrodoKEM、HQC、LAC、LEDAcrypt（merger of LEDAkem/LEDApkc）、NewHope、NTRU（merger of NTRUEncrypt/NTRU-HRSS-KEM）、NTRU Prime、NTS-KEM、ROLLO（merger of LAKE/LOCKER/Ouroboros-R）、Round5（merger of Hila5/Round2）、RQC、SABER、SIKE、Three Bears。第二轮的 9 名数字签名候选人是：CRYSTALS-DILITHIUM、FALCON、GeMSS、LUOV、MQDSS、Picnic、qTESLA、Rainbow、SPHINCS+。

思考 1.5：你所知道的近 5 年来的密码学大事件有哪些？ □

1.3 对加密体制的攻击和安全性*

对于加密体制，根据敌手具有的能力（或攻击方式）由弱到强分为：

（1）唯密文攻击（Ciphertext Only Attack，COA）。敌手（或密码分析者）只能通过考察密文来试图推导解密密钥或明文。

（2）已知明文攻击（Known Plaintext Attack，KPA）。敌手拥有一定量的明文和相对应的密文。

（3）选择明文攻击（Chosen Plaintext Attack，CPA1）。敌手可以选择明文，接着得到相对应的密文。之后，敌手使用所拥有的信息，恢复以前未见过的密文的相应明文。

（4）自适应选择明文攻击（Adaptive Chosen Plaintext Attack，CPA2）。一种选择明文攻击的特殊情况，在攻击中能以一种自适应的方式选择明文（来得到相应的密文），即明文的选择可依赖于前面已得到的密文。

（5）选择密文攻击（Chosen Ciphertext Attack，CCA1）。敌手可以选择密文，接着得到相应的明文。实现这种攻击的一种方法是敌手设法获取了解密设备的访问权（但不是解密密钥，因为解密密钥可能被安全地嵌入到设备中）。然后，在不访问该解密设备的情况下推导出（先前未询问过解密设备的）密文对应的明文。

（6）自适应选择密文攻击（Adaptive Chosen Ciphertext Attack，CCA2）。一种选择密文攻击的特殊情况，在攻击中能以一种自适应的方式选择密文（来得到相应的明文），即密文的选择可依赖于前面已得到的明文。

思考 1.6：上述 6 种攻击的区别是什么？

攻击者拥有的能力有所不同。在唯密文攻击中，攻击者不拥有任何预言机（Oracle，也叫谕示机），即任何可以“回答”攻击者构造的“询问”的机器，只能就现有的一些信息开展分析和攻击。选择明文攻击和自适应选择明文攻击的攻击者拥有加密预言机，该机器会“回答”攻击者构造的“询问”明文，给出相应的密文。而在选择密文攻击和自适应选择密文攻击中，攻击者拥有解密预言机，该机器会“回答”攻击者构造的“询问”密文，给出相应的明文。

初学者通常会对这个假设感到不解：为什么攻击者会获得预言机？

关于攻击者拥有预言机的假设是合理的，比如，攻击者获得了一台加密机，该加密机可以对攻击者输入的任意明文产生密文。或者，攻击者知道密文来自于两个明文之一，例如来自对“Yes”或者“No”的加密。后面我们将会学到，对于公钥密码系统，攻击者获得加密预言机是容易的（平凡的），因为用来加密的公钥是公开的，任何人都可以对想要加密的明文进行加密以得到相应的密文。 □

对加密体制而言，攻击的最终目标是得到明文，但如果能得到密钥，则必然可以得到明文。加密体制的安全性从低到高主要有以下三类。

（1）完全攻破：攻击者能得到使用的密钥（对公钥系统而言指私钥）。

（2）部分攻破：攻击者可能不需要知道密钥，而对某些密文能推导出明文。

（3）密文可区分：攻击者能以超过 1/2 的概率解决以下用两种不同形式描述的问题。一是给攻击者任意两个明文和其中任意一个明文的密文，攻击者判断该密文是哪个明文的密文；二是给攻击者任意一个明文和该明文对应的密文，以及一个和该密文等长的随机字符串，让攻击者判断哪个是密文。 □

思考 1.7：密文区分性为什么重要？

用户购物时通常明文就两种形式：“买”或者“不买”。军队打仗时，明文可能就是“攻”或者“不攻”。假设明文就一个比特，一个明文是 0，一个明文是 1。当面对这两个明文的密文时，攻击者即使知道这个密文来自于对 0 或者 1 的加密，也无法知道到底明文是 0 还是 1，即如果攻击者只能随机猜测（类似投硬币猜测），猜测成功的概率是 1/2，那么，加密体制就是密文不可区分的。如果攻击者猜对的概率超过 1/2，则说明攻击者不是随机猜测，而是“有优势”去猜测，那么，加密体制就是密文可区分的。 □

理论证明表明：密文不可区分（Indistinguishability）等价于语义安全（Semantic Security）。语义安全是指有限计算能力的敌手不能从密文中得到任何明文信息（与完善保密性相对应，后者是指无限计算能力的敌手不能从密文得到任何明文信息）。目前，加密体制的最好安全标准是适应性选择密文攻击条件下的密文不可区分性，记为 IND-CCA2。有兴趣的读者可以进一步深入学习，这是可证明安全理论的重要内容。

小结

本章给出了密码学的目标和主要知识构成。回顾了密码学的发展历史。还给出了对加密体制的攻击和安全性的基本概念。

本章重点是密码学的目标与知识构成，本章的难点是对加密体制的攻击。

扩展阅读建议

[1] RIVEST R. Cryptography and Security [OL]. http://people. csail. mit. edu/rivest/crypto-security. html.

[2] Cryptology pointers [OL]. http://research. cyber. ee/~lipmaa/crypto/.

习题

（1）举例说明攻击的种类和密码学的目标。

（2）信息安全的目标有哪些？给出英文名词及其解释。

（3）密码体制安全性有几种，分别是什么，它们有何差异？

（4）密码学历史中你认为重要的事件是什么？从这个历史演变中你觉得有何启示？

（5）分析密码学的发展简史，探讨其中的发展逻辑。

（6）撰写报告，说明近 5 年密码学的重大事件。

（7）对加密体制的攻击从敌手具有的能力来看分为哪些？

（8）对加密体制的攻破包括哪几种情况？

第 2 章　古典密码体制

本章要点：

- 古典加密方案，包括置换密码和代换密码。
- 代换密码，包括单表加密和多表加密。
- 唯密文攻击，包括：单表代换的统计（频率）分析攻击方法、多表代换加密的统计分析攻击方法。

古典密码体制，也称为传统密码体制或经典密码体制，主要通过字符间的置换和代换来实现。介绍古典密码体制的目的是提供密码设计和密码分析的基本案例，便于初学者入手。1949 年，Shannon 发表《保密系统的通信理论》一文之后，密码学从艺术变成了科学，此后的密码学称为现代密码学。

2.1　密码系统的概念模型

密码学最初的目的是保密。为了对加密系统有一个整体全貌，这里先给出密码系统的基本概念。

密码系统的模型如图 2.1 所示。

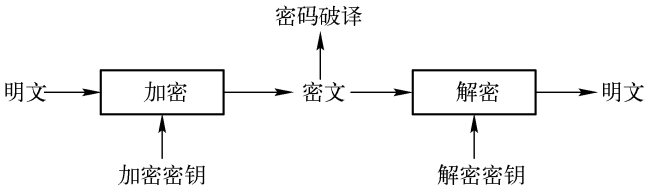


图 2.1　密码系统模型

定义 2.1　密码系统（Cryptosystem）

一个密码系统是一个五元组$\{P,C,K,E,D\}$，它满足：

（1）P 是可能明文的有限集（明文空间）。

（2）C 是可能密文的有限集（密文空间）。

（3）K 是一切可能密钥构成的有限集（密钥空间）。

（4）任意 $k\in K$，有一个加密算法 $e_{k_e}\in E$ 和相应的解密算法 $d_{k_d}\in D$，使得 $e_{k_e}:P\rightarrow C$ 和 $d_{k_d}:C\rightarrow P$ 分别为加密和解密函数，满足 $d_{k_d}(e_{k_e}(x))=x$，其中 $x\in P$。

注意：

（1）古典密码体制通常对字符进行运算，26 个英文字符常被抽象为 0~25 间的整数。（在计算机发明之后，现代密码体制通常对 bit 进行运算。）

（2）如果 $k_d=k_e$，即加密密钥和解密密钥相等，则称为对称密钥密码体制。否则，称为非对称密钥密码体制（又叫公钥密码体制）。古典方案都是对称密钥密码体制，公钥密码密

码体制是在 1976 年由 W. Diffie 和 M. Hellman 提出的，它的出现是密码学发展史上的一座里程碑。

（3）对称密码按照明文的类型可分为序列密码（又叫流密码，Stream Cipher）和分组密码（Block Cipher）。序列密码对明文按照字符或者比特逐位加密，对密文逐位解密。分组密码则将明文按照一定的长度分组（Block），加密和解密分组进行。（为说明两者间的关系，这里举个例子，序列密码可视为分组长度为 1 个比特或 1 个字节的分组密码。）公钥密码体制都是对分组进行运算的，故不再区分。

思考 2.1：古典密码系统模型中不涉及通信，与后面通信保密系统模型不同。 □

一个实用的密码系统需要有两个特性：

（1）易计算性。这是性能的主要方面，加密算法和解密算法应当“有效地”计算。即该密码系统是容易使用的。这里“有效地”概念需要用计算复杂性理论中的度量来解释。

（2）安全性。看到密文，无法知道相应的明文和密钥。这里只是给出朴素的对“安全性”的直观感受，严格的安全性定义主要分为两种：无条件安全和计算安全。对密码系统而言，无条件安全也称为完美安全（Perfect Security）或信息理论安全（Information Theoretical Security），即不对攻击者的计算能力作限定。计算安全则需要假设攻击者的计算能力。Shannon 最早从理论上研究了对称密码的安全性，第 3 章将详细介绍。

19 世纪末，A. Kerckhoffs 提出军事密码系统的设计原则，其中最重要的一条是：加密方案不应该保密，唯一需要保密的是密钥。也就是说，加密的安全性不应当依赖于加密方案的保密性，即永远不要假定敌手得不到加密方案。

2.2 置换加密体制

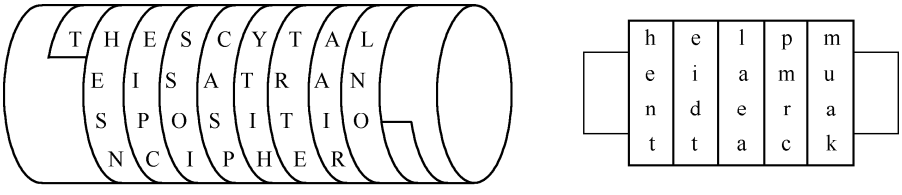
置换密码（Permutation Cipher），又称为换位密码（Transposition Cipher），指根据一定的规则重新排列明文。密文空间和明文空间相同，故密文只是打乱了明文字符的位置和次序。

例 2.1 一个通俗的例子，假设一副扑克牌是明文，那么，洗牌以后得到的就是密文。

例 2.2 Scytale 密码。

置换密码可以认为是最古老的一种加密方式，典型案例是古希腊斯巴达军队中使用的 Scytale 密码。Scytale 加密中沿着木棍写上明文字符，然后展开布条，即为密文，如图 2.2a 所示。加密的内容为“THE SCYTALE IS A TRANSPOSITION CIPHER”。

例如，假设明文是：“Help me I am under attack”（见图 2.2b），加密时写为



a) b)

图 2.2 Scytale 加密

于是得到密文是“henteidtlaeapmrcmuak”。

定义 2.2 设 S 是一个有限集合，ϕ 是从 S 到 S 的一个映射，如果对任意 $u,v \in S$，当 $u \neq v$ 时，$\phi(u) \neq \phi(v)$，则称 ϕ 为 S 上的一个置换（Permutation）。

定义 2.3 设 n 为一固定整数，P、C 和 K 分别为明文空间、密文空间和密钥空间。明/密文是长度为 n 的字符序列，分别记为 $X=\{x_1,x_2,\cdots,x_n\} \in P$，$Y=\{y_1,y_2,\cdots,y_n\} \in C$，$K$ 是定义在 $\{1,2,\cdots,n\}$ 内的所有置换的集合。对任何一个密钥 $\sigma \in K$（即一个置换），定义置换加密如下：

$$e_\sigma(x_1,x_2,\cdots,x_n)=(x_{\sigma(1)},x_{\sigma(2)},\cdots,x_{\sigma(n)})$$

$$d_{\sigma^{-1}}(y_1,y_2,\cdots,y_n)=(y_{\sigma^{-1}(1)},y_{\sigma^{-1}(2)},\cdots,y_{\sigma^{-1}(n)})$$

其中，σ^{-1} 是 σ 的逆置换，密钥空间 K 的大小为 $n!$。

显然，置换加密只是重新排列了明文的位置。可视为加密函数为空（NULL），密钥等于置换 σ。

例 2.3 设有限集 $X=\{1,2,3,4,5,6,7\}$，σ 为 X 上的置换，有 $\sigma(1)=3$，$\sigma(2)=1$，$\sigma(3)=5$，$\sigma(4)=2$，$\sigma(5)=4$，$\sigma(6)=7$，$\sigma(7)=6$。可直观表示为

$$\sigma=\begin{bmatrix}1&2&3&4&5&6&7\\3&1&5&2&4&7&6\end{bmatrix}$$

思考 2.2：请给出解密置换或者解密密钥 σ^{-1}。

结果为 $\sigma^{-1}=\begin{bmatrix}1&2&3&4&5&6&7\\2&4&1&5&3&7&6\end{bmatrix}$。 □

思考 2.3：多次置换（例如 σ^2）是否能够加强保密性？

不能。 □

2.3 代换加密体制

代换密码（Substitution Cipher）是将明文中的字符替换成其他字符的密码体制。基本方法是：建立一个代换表，加密时将待加密的明文字符通过查表代换为对应的密文字符，这个代换表就是密钥。

代换是密码体制中基本的处理技巧，它在分组密码（第 5 章介绍）设计中有广泛的应用。

例 2.4 一个通俗的例子，假设有一本汉字字典，可以将其视为一个代换表，通过查表进行加密。例如，某个明文字的密文字是该明文字的下一个汉字。

定义 2.4 设 P、C 和 K 分别为明文空间、密文空间和密钥空间。其中 P 和 C 是 26 个英文字母的集合，K 是由 26 个数字 0，1，…，25 的所有代换组成，对任何一个密钥 $\pi \in K$（即代换表），定义代换加密如下：

$$e_\pi(x)=\pi(x)$$

$$d_\pi(y)=\pi^{-1}(y)$$

其中，π^{-1} 表示密钥 π 的逆代换；x 表示一个明文字符；y 表示一个密文字符。

按照一个明文字符是否总是被一个固定的字母代换，可将代换密码划分为两类：单表代换密

码（Monoalphabetic Substitution Cipher）和多表代换密码（Polyalphabetic Substitution Cipher）。

2.3.1 单表代换密码

单表代换密码指明文中相同的字母，在加密时都使用同一个字母来代换。单表代换密码又分为移位密码（Shift Cipher）和仿射密码（Affine Cipher）。

1. 移位密码

将26个英文字符从a~z依次分别与0~25的整数建立一一对应关系。令 $P=C=K=Z_{26}$，$x\in P$，$y\in C$，$k\in K$，定义：

$$e_k(x)=x+k \bmod 26$$
$$d_k(y)=y-k \bmod 26$$

例2.5 凯撒（Caesar）密码。

Caesar密码是 $k=3$ 的移位密码，代换表如下：

π：	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
π^{-1}：	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

若明文为please，则密文为sohdvh。

由于该密码体制密钥量为26，仅有26种可能的密钥，故可以通过穷举法进行密码分析。

2. 仿射密码

移位密码的密钥量太小，且代换表中字母先后次序没有改变。仿射密码可以改进这些缺点。

令 $P=C=Z_{26}$，$K=\{(a,b)\in Z_{26}\times Z_{26}:\gcd(a,26)=1\}$，$x\in P$，$y\in C$，$k=(a,b)\in K$，定义：

$$e_k(x)=ax+b \bmod 26$$
$$d_k(y)=a^{-1}(y-b) \bmod 26$$

这里要求 $\gcd(a,26)=1$，否则，加密函数就不是一个单射函数。例如，当 $k=(6,1)$ 时，$\gcd(a,26)=\gcd(6,26)=2$，对 $x\in Z_{26}$，有 $6(x+13)+1=6x+1 \bmod 26$，于是 x 和 $x+13$ 都是 $6x+1$ 的明文。

思考2.4：证明 $\gcd(a,26)=1$ 时，仿射密码的解唯一。

证明如下：设存在 x_1，$x_2\in Z_{26}$，使得 $e_k(x)=ax_1+b=ax_2+b \bmod 26$，于是 $ax_1=ax_2 \bmod 26$，有 $26\mid a(x_1-x_2)$，又因为 $\gcd(a,26)=1$，所以 $26\mid(x_1-x_2)$，由于 $x_1,x_2\in Z_{26}$，得到 $x_1=x_2$。 □

思考2.5：仿射密码的密钥空间大小是多少？

a 的可能性为12，因为 $a\in Z_{26}$，$\gcd(a,26)=1$，即 $a=\phi(26)=\phi(2\times13)=\phi(2)\times\phi(13)=1\times12=12$。$b\in Z_{26}$，$b$ 的可能性为26种。故整个密钥空间大小为 $12\times26=312$。 □

思考2.6：移位密码是仿射密码的特例吗？

显然，当 $a=1$ 时，仿射密码就退化为移位密码。 □

3. 单表代换的安全性

单表代换存在的一个问题就是**密文和明文间有着固定的代换关系。**简单地说，明文字符

出现的频率没有被掩盖，即明文中常出现的字符在密文中也常出现，于是被密码分析者所利用，从而达到破解的目的。

在单表代换下，字母的频率、重复字母的模式和字母组合方式等统计特性（除了字母名称改变以外）均未改变。

表 2.1 给出了英文字母的出现频率统计。

表 2.1　26 个英文字母的出现频率统计（Letter Frequency）

字母	频率	字母	频率	字母	频率	字母	频率
a	8.167%	h	6.094%	o	7.507%	v	0.978%
b	1.492%	i	6.966%	p	1.929%	w	2.360%
c	2.782%	j	0.153%	q	0.095%	x	0.150%
d	4.253%	k	0.772%	r	5.987%	y	1.974%
e	12.702%	l	4.025%	s	6.327%	z	0.074%
f	2.228%	m	2.406%	t	9.056%		
g	2.015%	n	6.749%	u	2.758%		

一般来说，e 是频率最高的字母，t 排在第二位，a 或者 o 排在第三位，e、t、a、o、n、i、s、r、h 比任何其他字母有高得多的频率，约占英文文本的 70%。如果还考虑位置特征，字母 a、i、h 不常作为单词的结尾，而 e、n、r 出现在开始位置的情况比结尾位置的少，t、o、s 的出现在前后基本相等。

于是，密文分析时，可先统计密文中各个字母出现的频率，然后猜测出现次数最多的字母为 e 的密文，次多的字母为 t，或者 a，或者 o。然后用这两个字母来替换密文，根据搭配关系、词首尾关系继续分析，猜测剩余的密文，最终得到全部的明文。

2.3.2　多表代换密码

为了使密文中不表现出明文的统计特征，一个办法就是使每个明文字母可能被多种密文字母所代换，于是可以通过使用不止一个代换表进行代换来实现。

多表代换密码就是多个代换表依次对明文消息的字母进行代换的加密方法。

设明文序列为 $m=m_1,m_2,\cdots$，代换表序列为 $\pi=\pi_1,\pi_2,\cdots,\pi_d,\pi_1,\pi_2,\cdots,\pi_d,\cdots$，于是，得到的密文序列为

$$c=\pi(m)=\pi_1(m_1),\pi_2(m_2),\cdots,\pi_d(m_d),\pi_1(m_{d+1}),\pi_2(m_{d+2}),\cdots,\pi_d(m_{2d}),\cdots$$

通俗地说，就是代换表的个数为 d，用不同的代换表来轮流代换明文消息中的字母。

严格来说，这里 d 表示代换序列的周期。

思考 2.7：$d=1$ 时是什么情况，$d=\infty$ 是什么情况？

如果 $d=1$，多表代换密码退化为单表代换密码，即只有一个代换表。如果 $d=\infty$，则代换序列是非周期无限序列，每个明文都采用不同的代换表进行加密，这其实就是“一次一密”。通常，实际应用中采用周期代换密码，也就是使用有限个代换表。 □

多表代换密码利用从明文字母到密文字母的多个映射，隐藏了单字母的统计特征（如频率特征）。它将明文字母划分为长度相同的明文分组，然后对明文分组进行代换。这样，

同一个字母只要位于不同明文分组中的不同位置就会映射到不同的密文字母，从而更好地抵抗密码分析。

多表代换密码体制有 Playfair、Vigenere、Vernam、Hill、Beaufor、Running-Key、转轮机(Rotor Machine)。下面按照年代顺序依次介绍比较典型的 4 种：Playfair 密码、Vigenere 密码、Vernam 密码和 Hill 密码。转轮机由于是机械加密单独介绍。

1. Playfair 密码

Playfair 密码将明文按两个字母分成一组，每组根据代换表代换成其他两个字母。这里的代换表是一个 5×5 字母矩阵（I 和 J 等价，于是 26 个英文字母均在字母矩阵中），该矩阵可根据密钥构造，这里的密钥通常是一个关键词，假设如“hello”。构造矩阵的方法是：从左到右，从上到下依次填入关键词的字母，关键词中的重复字母第二次出现时略过，然后将剩余字母按顺序填入矩阵。约定表中第一列视为第五列的右边一列，第一行视为第五行的下一行。

每一对明文字母 p_1、p_2 分别替代成 c_1、c_2 的方法如下：

① 如果 p_1、p_2 同行，则 c_1、c_2 为紧靠 p_1、p_2 右端的字母。

② 如果 p_1、p_2 同列，则 c_1、c_2 为紧靠 p_1、p_2 下端的字母。

③ 如果 p_1、p_2 在不同行不同列，则确定了两个角，于是剩下的两个角为 c_1、c_2，按照同行的原则对应。

④ 如果 p_1、p_2 相同，则插入实现约定好的字母，并用上述方法处理。

⑤ 如果明文字母数为奇数，则在明文末尾添加一个事先约定好的字母将其填充为偶数。

解密时，则同样将密文按两个字母分为一组，用矩阵解密，只不过相应地将右端换成左端，下端换成上端。

例 2.6 如果密钥为“hello”，如何用 Playfair 密码加密“university”。

先构造字母矩阵如下：

$$\begin{bmatrix} h & e & l & o & a \\ b & c & d & f & g \\ i/j & k & m & n & p \\ q & r & s & t & u \\ v & w & x & y & z \end{bmatrix}$$

然后，明文分组为：un，iv，er，si，ty。加密结果为：tp，qh，cw，qm，yo。

2. Vigenere 密码

Vigenere 密码是典型的多表代换，加密中一个字母可被映射到 m 个可能的字母之一(假定密钥包括 m 个不同的字母)，所以分析起来比单表代换更困难。

定义 2.5 设 m 为某个固定的正整数，P、C、K 分别为明文空间、密文空间、密钥空间，且 $P=C=K=(Z_{26})^m$，对一个密钥 $k=(k_1,k_2,\cdots,k_m)$，定义：

$$e_k(x_1,x_2,\cdots,x_m)=(x_1+k_1,x_2+k_2,\cdots,x_m+k_m)$$

$$d_k(y_1,y_2,\cdots,y_m)=(y_1-k_1,y_2-k_2,\cdots,y_m-k_m)$$

其中，$(x_1,x_2,\cdots,x_m)$为一个明文分组中的 m 个字母。所有运算在 Z_{26} 中进行。密钥长度为 m，故密钥空间为 26^m。明文是按照长度为 m 的分组进行加密的。

例 2.7 设 $m=5$，密钥字为“hello”，如何加密“university”。

明文分组为：unive，rsity。密文转化为 Z_{26} 为：20，13，8，21，4 和 17，18，8，19，24。

密钥字实际为 $k=(7,4,11,11,14)$，Z_{26} 表示的密文为：1，17，19，6，18 和 24，22，19，4，12。密文字母为“brxgs，ywtem”。

3. Vernam 密码

Vernam 密码提出后一直被认为是不可破译的，但直到 1949 年才由 Shannon 给予了理论证明（详见第 3.3 节）。

定义 2.6 设加密的明文为 $p=p_1p_2\cdots p_i\cdots$，密钥为 $k=k_1k_2\cdots k_i\cdots$，其中 $p_i,k_i\in \mathrm{GF}(2)$，$i\geqslant 1$，则密文 $c=c_1c_2\cdots c_i\cdots$ 满足：

$$c_i=m_i\oplus k_i,i\geqslant 1$$

其中，$\oplus$ 为模 2 加法。

Vernam 密码无法经受已知明文攻击，这是因为 $k_i=m_i\oplus c_i,i\geqslant 1$，只要知道了某些明文-密文对，便可以迅速确定相应的密钥。如果同一个密钥被重复使用，密码分析者就可以立即解密密文得到明文。

为了避免密钥本身的重复，一种极端情况是：①密钥是真正的随机序列；②密钥至少和明文一样长；③一个密钥只使用一次。如果这样，密码就是不可破译的，这便是著名的“一次一密”（one time pad）。然而“一次一密”在实际上是行不通的，需要经常产生、存储、安全传递大量的很长的密钥，这在实际中是极其困难的。

4. Hill 密码

Hill 密码的基本思想是把 m 个连续的明文字母代换成 m 个连续的密文字母，这个代换由密钥决定，这个密钥是一个变换矩阵，解密时只需要对密文做一次逆变换即可。

定义 2.7 设 m 是某个固定的正整数，P、C、K 分别为明文空间、密文空间、密钥空间，且 $P=C=(Z_{26})^m$，设 $\boldsymbol{K}=(k_{ij})_{m\times m}$ 是一个 $m\times m$ 可逆矩阵，即行列式 $\det(\boldsymbol{K})\neq 0$，且 $\gcd(26,\det(\boldsymbol{K}))=1$。对任意密钥 $k\in\boldsymbol{K}$，定义：

$$e_k(x)=x\boldsymbol{K}$$

$$d_k(y)=y\boldsymbol{K}^{-1}$$

所有运算均在 Z_{26} 中进行。

思考 2.8：Hill 密码和仿射密码有何关系？

当 $m=1$ 时，Hill 密码退化成单字母仿射密码（移位密码）。 □

例 2.8 设 $m=2$，密钥 $\boldsymbol{K}=\begin{bmatrix}11 & 8\\ 3 & 7\end{bmatrix}$，设明文为“hill”，相应的明文向量为[7,8]和[11,11]，于是，相应的密文向量分别为

$$[7,8]\begin{bmatrix}11 & 8\\ 3 & 7\end{bmatrix}=[77+24,56+56]=[23,8],$$

$$[11,11]\begin{bmatrix}11 & 8\\ 3 & 7\end{bmatrix}=[121+33,88+77]=[24,9],$$

故密文为“xiyj”。

与 Playfair 密码相比，Hill 密码的分组长度可以更长（即大于 2），由矩阵的维度决定。与 Vigenere 密码相比，Hill 密码更加安全。（Hill 密码中某个明文字母对应的密文字母既与

加密矩阵相关，也与该分组其他字母相关，而 Vigenere 密码中，明文字母只与加密密钥相关。）

思考 2.9：Hill 密码的安全性如何？

Hill 密码能够较好地抵抗统计分析，抗唯密文攻击（COA）的强度较高，但是，它易受到已知明文攻击（KPA）。如例 2.8 中，只要知道两组明文-密文对，则可以通过解方程组求出密钥，即设 $\boldsymbol{K}=\begin{bmatrix} a & b \\ c & d \end{bmatrix}$，解方程组 $7a+8c=23, 7b+8d=8, 11a+11c=24, 11b+11d=9$。

通常，m 维密钥需要 m 个明文-密文对（每个明文具有 m 个分组）即可求出。 □

2.3.3 多表代换密码的统计分析*

与单表代换密码相比，多表代换密码虽然能抵御简单的统计分析，但是，如果密码分析者能够**确定密钥的长度**，则多表代换密码的分析就可转变成单表代换密码的分析。

原因是：可将密文按照密钥的长度（假设长度为 m）划分，不妨设为 $c[1]=(c_1, c_2, \cdots, c_m)$，$c[2]=(c_{m+1}, c_{m+2}, \cdots, c_{2m})$，…，$c[t]=(c_{(t-1)m+1}, c_{(t-1)m+2}, \cdots, c_{tm})$。然后，分别分析 $(c_1, c_{m+1}, \cdots, c_{(t-1)m+1})$，$(c_2, c_{m+2}, \cdots, c_{(t-1)m+2}), \cdots$，即 $(c_i, c_{m+i}, \cdots, c_{(t-1)m+i})(i=1, \cdots, m)$ 组成的每一组均由单表代换加密而来。从而，再利用针对单表代换密码的方法进行攻击。

这里以分析 Vigenere 密码为例讲解，首先确定密钥长度，然后确定密钥的内容。确定密钥长度的方法主要有 Kasiski 测试法和重合指数（Index of Coincidence）法。

1. Kasiski 测试法

Kasiki 测试法于 1863 年由普鲁士军官 F. Kasiski 提出，其基本原理如下：

明文中如果两个相同的明文片段之间的距离是密钥长度的**倍数**的话，那么在密文中，这两个相同的明文片段所对应的密文片段一定是相同的（其原因很明显，因为加密的密钥是相同的）。反过来，如果密文中出现两个相同的字母片段，它们所对应的明文字母片段未必相同，但相同的可能性很大。于是，可以通过观察密文中重复出现的密文字母片段来估计密钥的长度，即找出相同密文片段的间隔数，然后求出的最大公因子就有可能是密钥的长度。

例如：假定一个密文中包含下列重复出现的片段，重复片段间的距离标在括号中：QAR（150）、RAT（42）、FET（10）、ROPY（81）、CDR（57）。由于 3 是 150、42、10、81、57 的因子分解中出现最多的因子，所以密文的周期最有可能是 3。

2. 重合指数法

重合指数法由 W. F. Friedman 在 1918 年提出，其论文为《重合指数及其在密码学中的应用》（*The Index of Coincidence and Its Applications in Cryptography*）。该文献是 1949 年以前最有影响的密码学文献。

首先，思考一个问题：如果一个文本是来自 26 个字母表的随机文本，则每个字母以 1/26 的概率出现，其中有两个字母相同的概率为 $(1/26)^2$，由于共有 26 个字母，故总概率为 $26\times(1/26)^2=1/26\approx 0.038$。由于英文文本与随机文本不同，26 个字母出现的概率分别为 $p_0, p_1, \cdots, p_{25}$（见表 2.1），找到两个相同字母的概率为 $p_0^2+p_1^2+\cdots+p_{25}^2\approx 0.065$，和随机文本区别很大。下面给出重合指数的定义。

定义 2.8 设某种语言由 n 个字母组成，每个字母 i 出现的概率为 p_i（$1\leqslant i\leqslant n$），重合

指数是指两个随机字母相同的概率，记为 IC = $\sum_{i=1}^{n} p_i^2$ 。

IC 的作用是主要是：①在单表代换情况下，明文和密文的 IC 是相同的（对英文而言，均为 0.065），而在多表代换情况下，密文的 IC 值较小（更像随机文本，极值为 0.038）。于是可用来判断密文是多表代换加密还是单表代换加密。②通过计算 IC 值，看是否接近 0.065，来分析多表代换加密的密钥长度。

思考 2.10：在有限的密文长度情况下，如何计算 IC 值？

设 $x=x_1x_2\cdots x_n$ 是长度为 n 的密文，其中 $a,b,\cdots,z$ 在 x 中出现的次数分别为 $f_0,f_1,\cdots,f_{25}$，显然，从 x 中任取两个元素共有 $C(n,2)$ 种方法，选取的两个元素同时为第 i 个字母的情况共有 $C(f_i,2)$ 种，$0\leqslant i\leqslant 25$，故 x 的 IC 值为

$$\mathrm{IC}=\frac{\sum_{i=0}^{25}(f_i(f_i-1))}{n(n-1)}$$

这样，很自然地得到估计多表代换密钥长度的方法。假设密钥长度为 $m=2,3,4,\cdots$，将密文分成长度为 m 的多个分组，每个分组中的第 1 个字母（或第 i 个字母，$1\leqslant i\leqslant m$）组成一个密文段 x，计算其 IC，若接近 0.065，则说明 m 是正确的，若接近 0.038，则 m 不正确。□

例如，表 2.2 给出密钥长度 $m=1\sim5$ 时，计算的 IC 值。

表 2.2　密钥长度 $m=1\sim5$ 时的 IC 值

密钥长度 m	IC（串 1）	IC（串 2）	IC（串 3）	IC（串 4）	IC（串 5）	平均 IC
1	0.043					0.043
2	0.046	0.041				0.044
3	0.044	0.051	0.048			0.048
4	0.043	0.041	0.046	0.041		0.043
5	0.062	0.067	0.067	0.061	0.071	0.066

当 $m=1$ 时，整个密文视为一个串，IC = 0.043 表明是多表代换加密。$m=2$ 时，有两个串，分别为每个分组的首字母和尾字母。$m=3$ 时，有 3 个串，分别为每个分组的首字母、中间字母和尾字母。依此类推。$m=5$ 时，重合指数最接近 0.065。

密钥长度确定后，下一个任务是确定密钥的具体内容，不妨设密钥为 $(k_1,k_2,\cdots,k_m)$。

使用前面的表示方法，密文中的 $(c_1,c_{m+1},\cdots,c_{(t-1)m+1})$，$(c_2,c_{m+2},\cdots,c_{(t-1)m+2})$ 等，即 $(c_i,c_{m+i},\cdots,c_{(t-1)m+i})(1\leqslant i\leqslant m)$ 组成的每一组均为单表代换加密。仔细观察，这个单表代换就是“移位密码”。$(c_1,c_{m+1},\cdots,c_{(t-1)m+1})$ 其实就是由密钥为 k_1 的“移位密码”加密而来，$(c_2,c_{m+2},\cdots,c_{(t-1)m+2})$ 就是由密钥为 k_2 的“移位密码”加密而来。依次类推。$(c_m,c_{2m},\cdots,c_{tm})$ 是由密钥为 k_m 的“移位密码”加密而来。于是，可以通过密文串 $(c_i,c_{m+i},\cdots,c_{(t-1)m+i})(1\leqslant i\leqslant m)$ 确定 k_i。

思考 2.11：如何确定“移位密码”的密钥？

这个问题与求“移位密码”的密钥的唯密文攻击相同。密钥可以视为一种“滑动偏移

值”。还是依靠重合指数来判断：求密文串［不妨设$(c_1,c_{m+1},\cdots,c_{(t-1)m+1})$］中字母 a,b,…,z 的出现次数$f_0,f_1,\cdots,f_{25}$。令 $n'=n/m$ 表示该串的长度。于是，26 个字母在密文中出现的概率依次为$f_0/n',f_1/n',\cdots,f_{25}/n'$。由于每个密文字母是由明文字母“滑动”$k_1$ 后而得，例如明文字母 a 对应的密文字母为 $a+k_1$，故明文的 IC 值的期望应该为

$$\mathrm{IC}=\sum_{i=0}^{25}\frac{p_i f_{((i+k_1)\bmod 26)}}{n'}$$

这个值应该接近 0.065。于是通过 k_1 遍历 0 到 25 依次计算，找到使得 IC 最接近 0.065 的那个 k_1，从而确定 k_1。其他密钥的确定方法一样，只是分析的密文串不同。通常，k_i 从密文串$(c_i,c_{m+i},\cdots,c_{(t-1)m+i})(1\leqslant i\leqslant m)$中分析并确定。

表 2.3 给出了由重合指数测试得到的数据示意表，偏移 $k=0,\cdots,25$ 时，分别计算相应的 IC 值，为节省篇幅，部分数据略去。从最接近 0.065 的情况可知密钥为(2,7,4,1,24)。

表 2.3　重合指数测试数据示意表

密钥 k	IC（串 1）	IC（串 2）	IC（串 3）	IC（串 4）	IC（串 5）
0	0.036234	0.046983	0.051021	0.04121	0.032192
1	0.038442	0.041256	0.049102	0.066251	0.039402
2	0.062028	0.051258	0.048312	0.042316	0.035518
3	0.041232	0.041381	0.046465	0.044012	0.037823
4	0.046232	0.037120	0.066091	0.038202	0.042086
5	0.042624	0.042375	0.048212	0.031532	0.041025
6	0.044517	0.049512	0.047122	0.032952	0.033251
7	0.039711	0.067028	0.049102	0.041569	0.042102
⋮	⋮	⋮	⋮	⋮	⋮
23	0.041354	0.042822	0.042624	0.045652	0.044024
24	0.042467	0.044417	0.044417	0.043665	0.069123
25	0.045517	0.040411	0.039311	0.039514	0.042102

当然，如果知道一个明文-密文对，则很容易知道偏移值（即密钥）。因此，古典密码体制一般只能承受唯密文攻击，而不能承受已知明文攻击。 □

2.3.4　转轮密码机

古典密码体制实际上可以分为人工加密和机械加密两种。从 19 世纪 20 年代开始，人们逐渐发明出各种机械加解密设备用来处理数据的加解密运算，最典型的设备是转轮密码机(Rotor Machine)。转轮密码机是由一组布线轮和转轮轴组成的灵巧复杂的机械装置，可以实现长周期的多表代换，且加密和解密的过程由机械自动快速完成。

1918 年，德国发明家 Arthur Scherbius 发明了名叫 ENIGMA 的转轮密码机，意为“谜”，后来被德国因为第二次世界大战而装备军队使用，又改进了基本设计。另一个著名的转轮密码机是美军的 Haglin 密码机，由瑞典的工程师 Haglin 发明，在第二次世界大战时被盟军广泛使用。同时，第二次世界大战期间日军的 PURPLE 也是转轮密码机。转轮密码机的使用极大地提高了加解密速度，同时抗攻击性能也有很大的提高，在第二次世界大战中有着广泛的应用，是密码学发展史上的一座里程碑。

转轮密码机由一个输入的键盘和一组转轮组成，每个转轮上有 26 个字母的输入引脚，和 26 个字母的输出引脚，输入输出关系由内部连线决定。以 3 转轮为例，从左到右分别为慢轮子、中轮子、快轮子（通过齿轮控制）。按下某一键时，键盘输入的明文电信号从慢轮子进入转轮密码机，轮子之间传递电信号，最后从快轮子输出密文。每次击键后，快轮子就转动一格，这样就改变了中轮子和快轮子之间的对应关系。两次连续按“A”键，得到的密文结果不一样，于是形成多表代换关系。图 2.3 给出了其原理的示意图，图 2.4 给出了一个实例。在首次击“A”键时，输出“E”，输出后快轮子转动一格，导致中轮子和快轮子的接触点变化，再通过内部连线，改变了输出。再次击“A”键，输出为“B”（快轮子与 24 对应的是 18，18 对应“B”）。

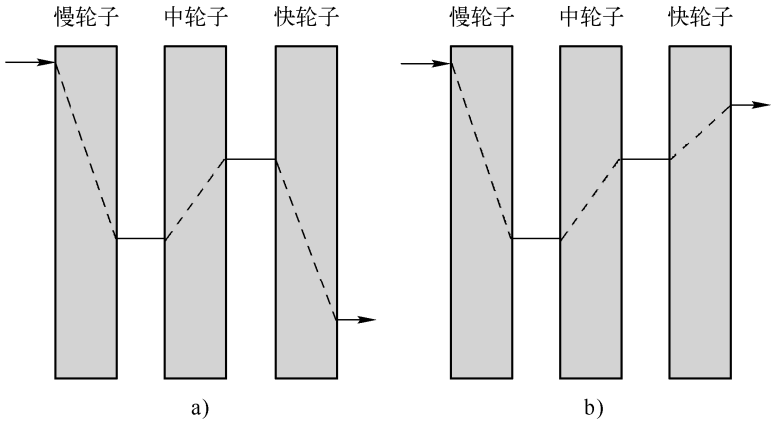


图 2.3　转轮密码机原理

a）击“A”键时的状态　b）击“A”键后的状态

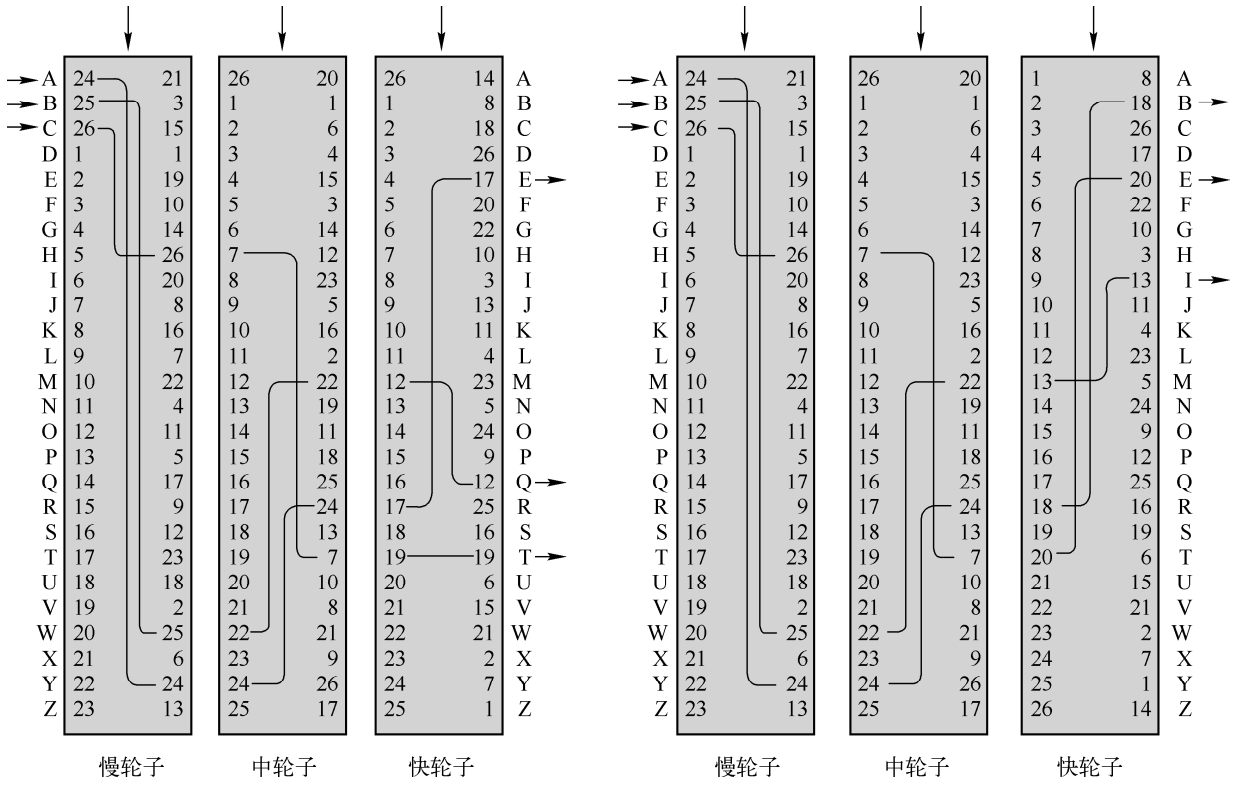


图 2.4　转轮密码机的一个实例

下面解释其原理。每个转轮相当于一个置换。多个转轮如果转速相同，则相当于一个转轮，还是一个置换。快轮子转动一圈（26 格），中轮子转动一格；中轮子转动一圈（26 格），慢轮子转动一格。由于多个转轮的转速不一样，转轮之间的对应关系在每次按键后均改变，于是形成多表代换。3 个转轮的密钥空间（周期）为 $26\times26\times26=17576$。通常，有 m 个转轮的密码机其密钥空间（周期）为 26^m。

另外，值得一提的是，由于转轮的旋转改变了字母的置换结果，通过频率统计的方法攻破 ENIGMA 行不通，但第二次世界大战期间阿兰·图灵（Alan Turning）参与了英国政府的破译行动，成功利用 ENIGMA 使用上的缺陷破译了密码，一定程度上改变了战争的局势。

思考 2.12：轮子的转动其实就是改变代换方式。如果把轮子的转动所导致的轮子间映射关系的改变视为代换（Substitution），把轮子内部的连线视为置换（Permutation），则转轮机其实就是后来用到的代换置换网络（SP 网络）的雏形（第 5 章 DES 分组密码的设计原理，第 4 章 Shannon 提出的设计密码的混淆思想和扩散思想）。多个转轮提高了安全性（增大了密钥空间），可能启发了 Shannon 提出乘积密码（第 3.5 节介绍）的雏形。 □

小结

本章首先介绍了密码系统的基本概念，然后介绍了置换加密体制，以及代换加密体制：包括单表代换密码和多表代换密码。单表代换密码又分为移位密码和仿射密码。多表代换介绍了典型的 4 种：Playfair 密码、Vigenere 密码、Vernam 密码和 Hill 密码。密码分析则介绍了针对单表代换的简单统计分析方法和针对多表代换的统计分析方法。古典密码学主要以人工操作字符为主，后期出现了机械操作。现代密码学是以计算机操作为主，主要对象是比特串。

本章知识要点总结如图 2.5 所示。

本章的重点是代换加密体制。本章的难点是多表代换加密体制的统计分析攻击，如重合指数法。

扩展阅读建议

关于古典密码学的最详尽的书籍是 David Kahn 于 1996 编写的 *The Codebreakers*：*The Comprehensive History of Secret Communication from Ancient Times to the Internet*。另外，作为课外阅读，可学习下 Hill 加密及其改进。

[1] HILL L S. Cryptography in an Algebraic Alphabet [J]. The American Mathematical Monthly, 1929, 36(6): 306-312.

[2] HILL L S. Concerning Certain Linear Transformation Apparatus of Cryptography [J]. The American Mathematical Monthly, 1931, 38(3): 135-154.

[3] OVERBEY J, TRAVES W, WOJDYLO J. On the Keyspace of the Hill Cipher [J]. Cryptologia. 2005. 29(1): 59-72.

[4] SAEEDNIA, SHAHROKH. How to Make the Hill Cipher Secure [J]. Cryptologia, 2000, 24(4): 353-360.

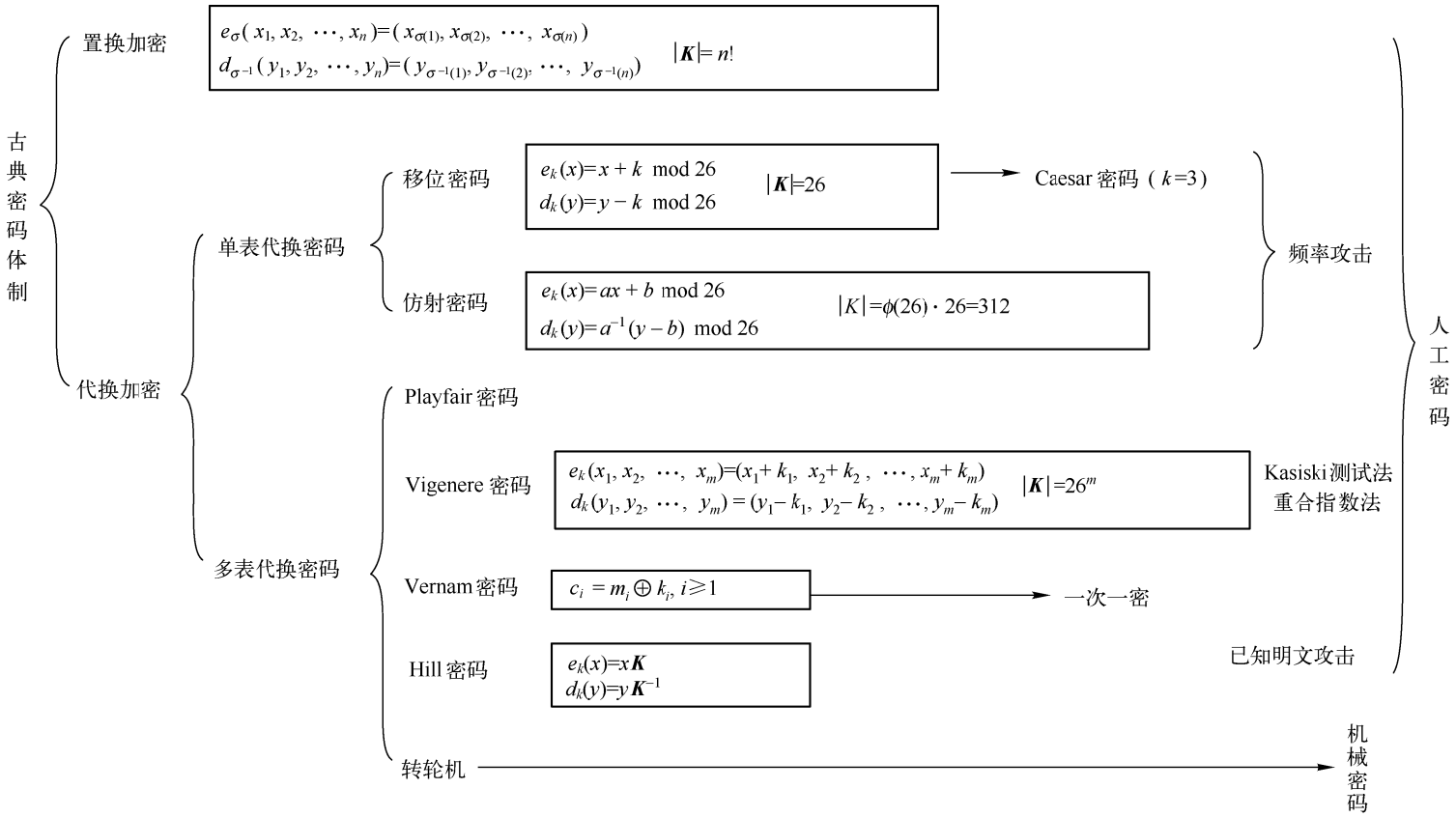

古典密码体制
置换加密
$e_\sigma(x_1, x_2, \cdots, x_n)=(x_{\sigma(1)}, x_{\sigma(2)}, \cdots, x_{\sigma(n)})$
$d_{\sigma^{-1}}(y_1, y_2, \cdots, y_n)=(y_{\sigma^{-1}(1)}, y_{\sigma^{-1}(2)}, \cdots, y_{\sigma^{-1}(n)})$
$|\boldsymbol{K}|=n!$
代换加密
单表代换密码
移位密码
$e_k(x)=x+k \mod 26$
$d_k(y)=y-k \mod 26$
$|\boldsymbol{K}|=26$
Caesar 密码（$k=3$）
仿射密码
$e_k(x)=ax+b \mod 26$
$d_k(y)=a^{-1}(y-b) \mod 26$
$|K|=\phi(26)\cdot 26=312$
频率攻击
多表代换密码
Playfair 密码
Vigenere 密码
$e_k(x_1, x_2, \cdots, x_m)=(x_1+k_1, x_2+k_2, \cdots, x_m+k_m)$
$d_k(y_1, y_2, \cdots, y_m)=(y_1-k_1, y_2-k_2, \cdots, y_m-k_m)$
$|\boldsymbol{K}|=26^m$
Kasiski测试法
重合指数法
Vernam 密码
$c_i=m_i \oplus k_i, i\geqslant 1$
一次一密
Hill 密码
$e_k(x)=x\boldsymbol{K}$
$d_k(y)=y\boldsymbol{K}^{-1}$
已知明文攻击
人工密码
转轮机
机械密码

图2.5　本章小结

上机实验

（1）编写程序演示几个典型的单表代换密码的工作过程。
（2）编写程序演示几个典型的多表代换密码的工作过程。

习题

（1）单表代换和多表代换的区别是什么？请各举出 3 例。
（2）单表代换的缺点是什么？
（3）一次一密为什么是不可破译的？
（4）多表代换分析的重点是什么？
（5）Kasiski 测试的主要目的是什么？其原理是什么？
（6）Hill 加密的抗选择明文攻击（CPA）的强度是多少，为什么？
（7）撰写报告，解释说明多表代换密码分析的方法。

第 3 章　信息理论安全

本章要点：

- 保密系统的数学模型，如何用信息论（概率论）表示保密系统。
- 完善保密性。
- 乘积密码。

Shannon 于 1948 年确立了现代信息论，并于 1949 年发表了《保密系统的通信理论》（Communication Theory of Secrecy Systems）一文，用概率统计的观点对信息保密问题做了全面的阐述。其贡献包括：

（1）以概率统计为工具对消息源、密钥源、接收和截取的消息进行数学描述和分析。

（2）用不确定性和唯一解距离度量了密码体制的安全性。

（3）给出了理论安全性的定义和充要条件。

（4）证明了一次一密的完善保密性。

（5）给出了实用密码设计的原则（扩散和混淆）。

（6）论述了多重密码。

Shannon 的论文使得信息论成为密码编码学和密码分析学的一个重要理论基础，宣告了科学的密码学时代的到来与信息论有关的一些基本概念，读者可参考本书附录。

3.1　保密系统的数学模型

随着通信的发展，1949 年 Shannon 在 *Bell Systems Technical Journal* 上发表了《保密系统的通信理论》一文，用概率统计的观点研究了信息的传输和保密问题。他认为通信系统的设计目的是在信道有干扰的情况下，使接收的信息无差错或差错尽可能小，如图 3.1 所示。保密系统的设计目的是使窃听者即使完全准确地接收到了信道上的传输信号也无法恢复原始信息，如图 3.2 所示。古典密码体制中看待加密系统是局部的静态的，而 Shannon 则用通信

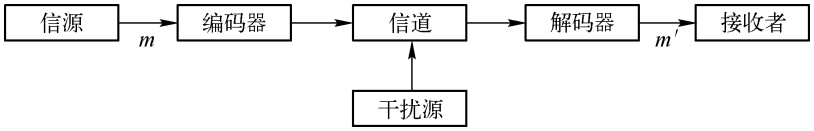


图 3.1　通信系统

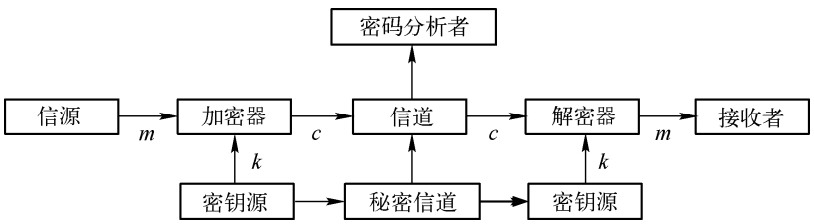


图 3.2　保密系统

的观点来看待保密系统。

1. 信源

在保密系统中，信源是信息的发送者。离散信源可以产生字符或字符串。设源字母表为：$X=\{a_i \mid i=0,1,\cdots,q-1\}$，其中 q 是一个正整数，表示信源中字母的个数。字母 a_i 出现的频率记为 $\Pr(a_i)$，$0\leqslant \Pr(a_i)\leqslant 1$，$0\leqslant i\leqslant q-1$，且 $\sum_{i=0}^{q-1}\Pr(a_i)=1$。如果只考虑长为 r 的信源，则明文空间为

$$M=\{m=(m_1,m_2,\cdots,m_i,\cdots,m_r) \mid m_i\in X,1\leqslant i\leqslant r\}。$$

如果信源是无记忆的，则

$$\Pr(m)=\Pr(m_1,m_2,\cdots,m_i,\cdots,m_r)=\prod_{i=1}^{r}\Pr(m_i)$$

如果信源是有记忆的，则需要考虑明文空间 M 中各元素的概率区别。信源的统计特性对密码体制的设计和分析有重要的影响。

2. 密钥源

密钥源用于产生密钥。密钥通常是离散的。设密钥源字母表为 $B=\{b_i \mid i=0,1,2,\cdots p-1\}$，其中 p 是一个正整数，表示密钥源字母表中字母的个数。字母 b_i 的出现概率记为 $\Pr(b_i)$，$0\leqslant \Pr(b_i)\leqslant 1$，$0\leqslant i\leqslant p-1$，且 $\sum_{i=0}^{p-1}\Pr(b_i)=1$。

密钥源通常是无记忆的，并且满足均匀分布。因此 $\Pr(b_i)=\dfrac{1}{p}$，$0\leqslant i\leqslant p-1$。如果只考虑长度为 s 的密钥，则密钥空间为

$$K=\{k=(k_1,k_2,\cdots,k_i,\cdots,k_s) \mid k_i\in B,1\leqslant i\leqslant s\}$$

一般而言，明文空间和密钥空间是相互独立的。合法的密文接收者知道密钥空间 K 和所使用的密钥 k。

3. 加密器

加密器用于将明文 $m=(m_1,m_2,\cdots,m_i,\cdots,m_r)$ 在密钥 $k=(k_1,k_2,\cdots,k_i,\cdots,k_s)$ 的控制下变换为密文 $c=(c_1,c_2,\cdots,c_t)$，即

$$(c_1,c_2,\cdots,c_t)=E_k(m_1,m_2,\cdots,m_i,\cdots,m_r)$$

其中，t 是密文的长度。所有可能的密文构成密文空间 C。设密文字母表为 Y，它是密文中出现的所有不同的字符的集合，则密文空间为

$$C=\underbrace{Y\times Y\times\cdots\times Y}_{t}$$

通常密文字母表与明文字母表相同，即 $Y=X$。一般而言，密文的长度与明文的长度也相同，即 $t=r$。

密文空间的统计特性由明文空间的统计特性和密钥空间的统计特性所决定。对任意的密钥 $k\in K$，则使用该密钥得到的密文集合为

$$C_k=\{E_k(m)\in C \mid m\in M\}$$

由于明文空间与密钥空间是相互独立的，所以对任意的 $c\in C$，有

$$\Pr(c)=\sum_{k\in\{k \mid c\in C_k\}}\Pr(k)\Pr(D_k(c)) \tag{3.1}$$

又因为

$$\Pr(c \mid m) = \sum_{k \in \{k \mid m = D_k(c)\}} \Pr(k) \tag{3.2}$$

所以根据贝叶斯（Bayes）公式，可得

$$\Pr(m \mid c) = \frac{\Pr(m)\Pr(c \mid m)}{\Pr(c)} = \frac{\Pr(m) \sum\limits_{k \in \{k \mid m = D_k(c)\}} \Pr(k)}{\sum\limits_{k \in \{k \mid c \in C_k\}} \Pr(k)\Pr(D_k(c))} \tag{3.3}$$

从式（3.1）~式（3.3）可以看出，知道明文空间和密钥空间的概率分布，就可以确定密文空间的概率分布 $\Pr(c)$，密文空间关于明文空间的概率分布 $\Pr(c \mid m)$，以及明文空间关于密文空间的概率分布 $\Pr(m \mid c)$。

为便于理解，可以进行一个直观的类比，如果把 c 视为终点，k 视为路径，m 视为起点，则图 3.3 就是人为构造的加密示意图，对式（3.1）~式（3.3）的解释如下：

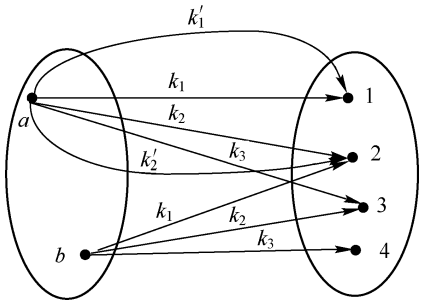


图 3.3　加密示意图

（1）密文集合有 $C_{k_1} = \{1,2\}$，$C_{k_2} = \{2,3\}$，$C_{k_3} = \{3,4\}$ 等。

（2）$\Pr(c)$ 为到达 c 点的概率，计算方法就是对某个连接 c 点的路径 k（即 $k \in \{k \mid c \in C_k\}$），寻找其起点 $D_k(c)$，该起点的概率为 $\Pr(D_k(c))$，该起点与终点间路径的概率 $\Pr(k)$，两者相乘，如此反复，最后求和。例如：

$$\Pr(c=1) = \Pr(a) \times \Pr(k_1') + \Pr(a) \times \Pr(k_1)$$

$$\Pr(c=2) = \Pr(a) \times \Pr(k_2') + \Pr(a) \times \Pr(k_2) + \Pr(b) \times \Pr(k_1)$$

（3）$\Pr(c \mid m)$ 为从起点 m 出发，到达终点 c 的概率。计算时，首先将起点和终点分别固定为 m 和 c，观察两者之间的路径 k 可能有多个（$k \in \{k \mid m = D_k(c)\}$），计算这样的路径概率 $\Pr(k)$ 之和。例如：

$$\Pr(c=1 \mid m=a) = \Pr(k_1) + \Pr(k_1')$$

$$\Pr(c=2 \mid m=a) = \Pr(k_2) + \Pr(k_2')$$

（4）$\Pr(m \mid c)$，需要用到 Bayes 公式。到达 c 点的起点可能有多个，求其中起点为 m 所占的比重。计算方法是：从 m 出发的概率（$\Pr(m)$），经过所有的路径 k，到达 c 点的概率（即 $\Pr(c \mid m)$），占整个到达 c 点概率的比重。

例如：

$$\begin{aligned}
\Pr(m=a \mid c=2) &= \frac{\Pr(m=a)\Pr(c=2 \mid m=a)}{\Pr(c=2)} \\
&= \frac{\Pr(m=a)\Pr(c=2 \mid m=a)}{\Pr(m=a)\Pr(c=2 \mid m=a) + \Pr(m=b)\Pr(c=2 \mid m=b)} \\
&= \frac{\Pr(a)[\Pr(k_2') + \Pr(k_2)]}{\Pr(b) \times \Pr(k_2) + \Pr(a)[\Pr(k_2') + \Pr(k_2)]}
\end{aligned}$$

下面给出一个正式的例子。

例 3.1　设有一个密码系统，明文空间 $M = \{a,b\}$，概率分布为 $\Pr(a) = 1/4$，$\Pr(b) =$

3/4。密钥空间 $K=\{k_1,k_2,k_3\}$，概率分布为 $\Pr(k_1)=1/2$，$\Pr(k_2)=1/4$，$\Pr(k_3)=1/4$。密文空间 $C=\{1,2,3,4\}$。加密变换为

$$E_{k_1}(a)=1,E_{k_1}(b)=2$$
$$E_{k_2}(a)=2,E_{k_2}(b)=3$$
$$E_{k_3}(a)=3,E_{k_3}(b)=4$$

计算 $H(M)$、$H(K)$、$H(C)$、$H(M\mid C)$、$H(K\mid C)$。

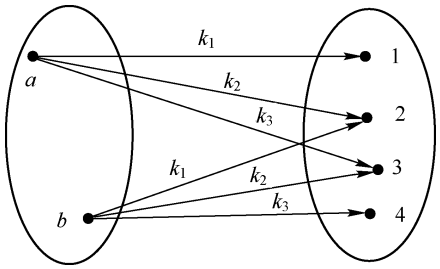


图 3.4　加密示意图

先绘制一个示意图，如图 3.4 所示。

$$\begin{aligned}H(M)&=-\Pr(a)\log_2\Pr(a)-\Pr(b)\log_2\Pr(b)\\&=-\frac{1}{4}\log_2\frac{1}{4}-\frac{3}{4}\log_2\frac{3}{4}\\&=-\frac{1}{4}\times(-2)-\frac{3}{4}(\log_2 3-2)\\&\approx 0.81\end{aligned}$$

$$\begin{aligned}H(K)&=-\Pr(k_1)\log_2\Pr(k_1)-\Pr(k_2)\log_2\Pr(k_2)-\Pr(k_3)\log_3\Pr(k_3)\\&=-\frac{1}{2}\log_2\frac{1}{2}-\frac{1}{4}\log_2\frac{1}{4}-\frac{1}{4}\log_2\frac{1}{4}\\&=\frac{3}{2}\end{aligned}$$

下面求 $H(C)$，需要先计算密文的概率分布：

$$\Pr(1)=\Pr(a)\Pr(k_1)=\frac{1}{4}\times\frac{1}{2}=\frac{1}{8}$$
$$\Pr(2)=\Pr(a)\Pr(k_2)+\Pr(b)\Pr(k_1)=\frac{1}{4}\times\frac{1}{4}+\frac{3}{4}\times\frac{1}{2}=\frac{7}{16}$$
$$\Pr(3)=\Pr(a)\Pr(k_3)+\Pr(b)\Pr(k_2)=\frac{1}{4}\times\frac{1}{4}+\frac{3}{4}\times\frac{1}{4}=\frac{1}{4}$$
$$\Pr(4)=\Pr(b)\Pr(k_3)=\frac{3}{4}\times\frac{1}{4}=\frac{3}{16}$$

于是

$$H(C)=-\frac{1}{8}\log_2\frac{1}{8}-\frac{7}{16}\log_2\frac{7}{16}-\frac{1}{4}\log_2\frac{1}{4}-\frac{3}{16}\log_2\frac{3}{16}\approx 1.85$$

下面计算 $H(M\mid C)$，需要首先计算已知密文情况下明文的概率分布：

$$\Pr(1\mid a)=\Pr(k_1)=\frac{1}{2},\quad \Pr(1\mid b)=0$$
$$\Pr(2\mid a)=\Pr(k_2)=\frac{1}{4},\quad \Pr(2\mid b)=\Pr(k_1)=\frac{1}{2}$$
$$\Pr(3\mid a)=\Pr(k_3)=\frac{1}{4},\quad \Pr(3\mid b)=\Pr(k_2)=\frac{1}{4}$$
$$\Pr(4\mid a)=0,\qquad \Pr(4\mid b)=\Pr(k_3)=\frac{1}{4}$$

由 Bayes 公式 $\Pr(x\mid y)=\dfrac{\Pr(x)\Pr(y\mid x)}{\Pr(y)}$，可计算得到

$$\Pr(a\mid 1)=\frac{\Pr(a)\Pr(1\mid a)}{\Pr(1)}=\frac{\frac{1}{4}\times\frac{1}{2}}{\frac{1}{8}}=1$$

同样可计算出 $\Pr(b\mid 1)=0$，$\Pr(a\mid 2)=1/7$，$\Pr(b\mid 2)=6/7$，$\Pr(a\mid 3)=1/4$，$\Pr(b\mid 3)=3/4$，$\Pr(a\mid 4)=0$，$\Pr(b\mid 4)=1$。

于是

$$\begin{aligned}H(M\mid C)&=\Pr(1)H(M\mid 1)+\Pr(2)H(M\mid 2)+\Pr(3)H(M\mid 3)+\Pr(4)H(M\mid 4)\\&=-1/8(1\times\log_2 1+0\times\log_2 0)-7/16\left(\frac{1}{7}\log_2\frac{1}{7}+\frac{6}{7}\times\log_2\frac{6}{7}\right)-\\&\quad 1/4\left(\frac{1}{4}\times\log_2\frac{1}{4}+\frac{3}{4}\times\log_2\frac{3}{4}\right)-3/16(0\times\log_2 0+1\times\log_2 1)\approx 0.46\end{aligned}$$

思考 3.1：如何计算 $H(K\mid C)$。

首先，计算

$$\Pr(1\mid k_1)=\Pr(a)=\frac{1}{4},\quad \Pr(1\mid k_2)=0,\quad \Pr(1\mid k_3)=0$$

$$\Pr(2\mid k_1)=\Pr(b)=\frac{3}{4},\quad \Pr(2\mid k_2)=\Pr(a)=\frac{1}{4},\quad \Pr(2\mid k_3)=0$$

$$\Pr(3\mid k_1)=0,\quad \Pr(3\mid k_2)=\Pr(b)=\frac{3}{4},\quad \Pr(3\mid k_3)=\Pr(a)=\frac{1}{4}$$

$$\Pr(4\mid k_1)=0,\quad \Pr(4\mid k_2)=0,\quad \Pr(4\mid k_3)=\Pr(b)=\frac{3}{4}$$

由 Bayes 公式可计算出

$$\Pr(k_1\mid 1)=1,\quad \Pr(k_2\mid 1)=0,\quad \Pr(k_3\mid 1)=0$$

$$\Pr(k_1\mid 2)=\frac{6}{7},\quad \Pr(k_2\mid 2)=\frac{1}{7},\quad \Pr(k_3\mid 2)=0$$

$$\Pr(k_1\mid 3)=0,\quad \Pr(k_2\mid 3)=\frac{3}{4},\quad \Pr(k_3\mid 3)=\frac{1}{4}$$

$$\Pr(k_1\mid 4)=0,\quad \Pr(k_2\mid 4)=0,\quad \Pr(k_3\mid 4)=1$$

于是

$$\begin{aligned}H(K\mid C)&=\Pr(1)H(K\mid 1)+\Pr(2)H(K\mid 2)+\Pr(3)H(K\mid 3)+\Pr(4)H(K\mid 4)\\&=-1/8(1\times\log_2 1+0\times\log_2 0+0\times\log_2 0)-\\&\quad 7/16\left(\frac{6}{7}\times\log_2\frac{6}{7}+\frac{1}{7}\times\log_2\frac{1}{7}+0\times\log_2 0\right)-\\&\quad 1/4\left(0\times\log_2 0+\frac{3}{4}\times\log_2\frac{3}{4}+\frac{1}{4}\times\log_2\frac{1}{4}\right)-\\&\quad 3/16(0\times\log_2 0+0\times\log_2 0+1\times\log_2 1)\\&\approx 0.46\end{aligned}$$

另外，在保密系统的研究中，通常假定信道是无干扰的，因此合法的密文接收者能够利用解密变换和密钥通过密文恢复明文，即 $m=D_k(c)=D_k(E_k(m))$。假定敌手可以从信道截获密文，敌手知道所用的密码体制，以及明文空间和密钥空间的统计特性，那么密码体制的安全性就完全取决于选用的密钥的安全性。 □

3.2 完善保密性

在保密系统中，信源是消息的发送者。密码设计者的努力方向是在设计密码体制时，尽可能地使破译者从密文中少获得明文的信息。根据各种熵之间的关系，结合密码系统 $F=(P,C,K,E,D)$，P 为明文空间，C 为密文空间，K 为密钥空间，E、D 分别为加密和解密函数。$H(P\mid C)$，即未被密文泄露的明文信息。已知密文条件下的密钥含糊度 $H(K\mid C)$，即未被密文泄露的密钥信息。

1. 已知密文条件下的密钥含糊度

定理 3.1 设 $F=(P,C,K,E,D)$ 为一个保密系统，则

$$H(K\mid C)=H(K)+H(P)-H(C)$$

证明：由熵的链法则，

$$H(K,P,C)=H(C\mid K,P)+H(K,P)=H(P\mid K,C)+H(K,C)。$$

由于密钥和明文唯一确定密文，密钥和密文唯一确定明文，密钥和明文统计独立，所以 $H(C\mid K,P)=H(P\mid K,C)=0$，$H(K,P)=H(K)+H(P)$。

从而 $H(K)+H(P)=H(K,C)$，又因为 $H(K,C)=H(C)+H(K\mid C)$，故 $H(K\mid C)=H(K)+H(P)-H(C)$。 □

2. 完善保密的定义（已知密文条件下的明文含糊度）

明文空间与密文空间的互信息为 $I(P;C)=H(P)-H(P\mid C)$，反映了密文空间所包含的明文空间的信息。因此，$I(P;C)$ 最小化是密码系统的一个重要设计目标。Shannon 定义了当攻击者具备无限计算资源时，在唯密文攻击下的安全性为完善保密性（Perfect Secrecy），并用他所创立的信息论中的熵和平均互信息的概念刻画了完善保密性，进而证明了一次一密具有完善保密性。

定义 3.1 设 $F=(P,C,K,E,D)$ 为一个保密系统，如果 $H(P\mid C)=H(P)$ 或者 $I(P;C)=0$，则 F 是完善保密的（又叫作无条件保密，信息理论安全）。

直观地说，从定义可知密文空间不包含明文空间的信息。观察到密文条件下明文的含糊度和明文本身的含糊度是一样的，即看到密文和没有看到密文对明文的含糊度来说效果是一样的。

通过概率来表述定义则可能更加直观：

定义 3.2 明文空间为 P 的加密方案是完善保密加密，若对 P 上任意的概率分布，任何明文 $p\in P$，任何密文 $c\in C$ 且 $\Pr(C=c)>0$，则有

$$\Pr(P=p\mid C=c)=\Pr(P=p)$$

看到密文并猜测其是某个明文的概率[$\Pr(P=p\mid C=c)$]和没有看到密文时明文本身的概率[$\Pr(P=p)$]，是一样的。即看到密文对明文的猜测并没有什么帮助，这再一次表述了密文空间不包含明文空间的信息。

从另一角度来看，$\Pr(P=p \mid C=c)$是一个使用 Bayes 公式求解的条件概率，

$$\Pr(P=p \mid C=c)=\frac{\Pr(P=p)\Pr(C=c \mid P=p)}{\Pr(c)}=\Pr(P=p)$$

于是说明 $\Pr(C=c)=\Pr(C=c \mid P=p)$。

3. 完善保密对密钥空间大小的要求

定理 3.2 $I(P;C)\geqslant H(P)-H(K)$。

证明：因为

$$H(P \mid C,K)=0,$$

$$\begin{aligned}H(P \mid C) &\leqslant H(P \mid C)+H(K \mid P,C)=H(P,K \mid C)\\ &=H(K \mid C)+H(P \mid C,K)=H(K \mid C)\leqslant H(K)\end{aligned}$$

这样，$H(P)-H(P \mid C)\geqslant H(P)-H(K)$，即 $I(P;C)\leqslant H(P)-H(K)$。

完善保密存在的必要条件是 $0=I(P;C)\leqslant H(P)-H(K)$，即 $H(P)\leqslant H(K)$。一般情形下，密钥空间满足均匀分布，因此 $H(K)=\log_2|K|$，$|K|$为密钥空间的大小。换句话说，系统密钥个数的对数必须不小于明文的熵。 □

4. 一次一密的完善保密性的证明

思考 3.2：如何证明一次一密具有完善保密性。 □

定理 3.3 一次一密具有完善保密性。

证明方法 1：假设加密长度为 L 的明文，有 26^L 个等可能的密钥，故 $H(K)=L\log_2 26$。因为对每个明文 $p_i\in P$ 和密文 $c_i\in C$，都有唯一一个密钥 $k_i\in K$，即

$$\Pr(C=c_i \mid P=p_i)=\frac{1}{26^L}$$

于是，$H(C \mid P)=L\log_2 26$。有 26^L 个等可能的密文，即 $\Pr(C=c_i)=1/26^L$，于是 $H(C)=L\log_2 26$。于是，$I(P;C)=H(C)-H(C \mid P)=0$。得证。

一个一般性的证明如下：因为 P,K,C 中已知其中两个就可确定第三个，且 P 与 K 独立。

$$H(P,K,C)=H(P,C)=H(P \mid C)+H(C)$$

$$H(P,K,C)=H(P,K)=H(P)+H(K)$$

联立以上两式，利用 $H(C)=H(K)$，得 $H(P \mid C)=H(P)$。 □

证明方法 2：利用概率表述的方法证明。$\Pr(C=c)=\Pr(C=c \mid P=p)$。在“一次一密”中，因为对每个明文 $p_i\in P$ 和密文 $c_i\in C$，都有唯一一个密钥 $k_i\in K$，即

$$\Pr(C=c_i \mid P=p_i)=\frac{1}{26^L}$$

$$\Pr(C=c_i)=\sum_{1}^{26^L}\Pr(C=C_i \mid P=p_i)\Pr(P=p_i)=26^L\times(1/26^L\times 1/26^L)=1/26^L$$

$$\Pr(C=c_i \mid P=p_i)=1/26^L$$

由 p_i，c_i 的任意性，有 $\Pr(C=c)=\Pr(C=c \mid P=p)$。 □

5. 完善保密的充分必要条件

定理 3.4 设 $F=(P,C,K,E,D)$是一个密码体制，且 $|M|=|C|=|K|$，则密码体制 F 具有完善保密性当且仅当密钥的选取满足均匀分布，并且对任意 $x\in P$ 和任意 $y\in C$，都存在唯一的密钥 $k\in K$，使得 $E_k(x)=y$。

证明：（1）首先，证明必要性。假设 F 具有完善保密性。固定一个 $x \in P$，对任意 $y \in C$，至少存在一个密钥 $k \in K$，使得 $E_k(x)=y$。因此，

$$|C| = |\{E_k(x) \mid k \in K\}| \leqslant |K|$$

因为假设 $|C| = |K|$，所以一定有：

$$|\{E_k(x) \mid k \in K\}| = |K|$$

也就是说，不存在两个密钥 $k_1 \in K$，$k_2 \in K$，使得 $E_{k_1}(x)=E_{k_2}(x)$，因此，对任意 $x \in P$ 和任意 $y \in C$，存在为唯一的密钥 $k \in K$，使得 $E_k(x)=y$。

令 $n=|K|$，$P=\{x_1,x_2,\cdots,x_n\}$，固定一个 $y \in C$，令 k_i 是满足 $E_{k_i}(x_i)=y$ 的密钥，$1 \leqslant i \leqslant n$，由 Bayes 定理知，

$$\Pr(x_i \mid y)=\frac{\Pr(y \mid x_i)\Pr(x_i)}{\Pr(y)}=\frac{\Pr(k_i)\Pr(x_i)}{\Pr(y)}$$

因为 F 具有完善保密性，故 $\Pr(x_i \mid y)=\Pr(x_i)$。代入上式，有 $\Pr(k_i)=\Pr(y)$，$1 \leqslant i \leqslant n$。说明密钥的选取是等概率的，因此，一定有：

$$\Pr(k_i)=1/|K|, 1 \leqslant i \leqslant n$$

（2）然后，证明充分性。注意到明文与密文是相互独立的，因此，对任意 $y \in C$，有

$$\begin{aligned}\Pr(y) &= \sum_{k \in K}\Pr(k)\Pr(D_k(y)) \\ &= \frac{1}{n}\sum_{k \in K}\Pr(D_k(y))\end{aligned}$$

其中，n 是不同的密钥的个数，因为 $\{D_{k_1}(y),D_{k_2}(y),\cdots,D_{k_n}(y)\}$ 是 $P=\{x_1,x_2,\cdots,x_n\}$ 的一个重新排列，故 $\sum_{k \in K}\Pr(D_k(y)) = \sum_{i=1}^{n}\Pr(x_i) = 1$。因此，对任意 $y \in C$，$\Pr(y)=1/n$。另外，对任意 $x \in P, y \in C$，令 $k \in K$ 是满足 $E_k(x)=y$ 的唯一密钥，则 $\Pr(y \mid x)=\Pr(k)=1/n$。于是，对任意 $x \in P, y \in C$，有 $\Pr(y \mid x)=\Pr(y)$，根据 Bayes 定理，可得 $\Pr(x \mid y)=\Pr(x)$。因此，密码体制 F 是完善保密的。 □

一次一密具有完善保密性，但该体制中真随机密钥字母的实现很困难，而且该体制所需密钥数量同明文数量一样，即随着明文的增长，密钥也会同步增长，从而给密钥的存储、传输和管理带来很大的难度。此外，接收方和发送方的同步也很难解决。总之，一次一密体制的成本很高，较难实现，不适合于广泛使用。不过，在外交和军事领域，一次一密体制仍然有着重要的用途。

3.3 乘积密码体制

Shannon 的另外一个贡献在于提出了设计密码的一些思想，如扩散和混淆，这些内容将在 5.1 节介绍。这里介绍乘积密码体制（Multiplicative Cipher）。即简单解释并回答这个问题：如果多个密码体制合并使用，形成“积”，是否可以加强密码体制的安全性？

设 $S_1=(P_1,C_1,K_1,E_1,D_1)$ 和 $S_2=(P_2,C_2,K_2,E_2,D_2)$ 是两个密码体制，S_1、S_2 定义了乘积密码体制：$(P_1 \times P_2, C_1 \times C_2, K_1 \times K_2, E_1 \times E_2, D_1 \times D_2)$，记为 $S_1 \times S_2$。在实际应用中，明文空间和密文空间通常都是相同的，即 $P_1=P_2=C_1=C_2$，于是乘积密码体制 $S_1 \times S_2$ 可以简化为（P，

$P, K_1 \times K_2, E, D)$，其中 $E = E_1 \times E_2$，$D = D_1 \times D_2$。

对任意的 $x \in P$，$k = (k_1, k_2) \in K_1 \times K_2$，加密变换为 $E_k(x) = E_{k_2}(E_{k_1}(x))$。对任意的 $y \in C$，$k = (k_1, k_2) \in K_1 \times K_2$，解密变换为 $D_k(y) = D_{k_1}(D_{k_2}(y))$。

显然，$D_k(E_k(x)) = D_k(E_{k_2}(E_{k_1}(x)) = D_{k_1}(D_{k_2}(E_{k_2}(E_{k_1}(x)))) = D_{k_1}(E_{k_1}(x)) = x$。

思考 3.3：Affine 密码可视为哪两种古典密码体制的乘积密码？

可视为移位密码和 Hill 密码（$m = 1$）的乘积密码。 □

对于乘积密码体制中密钥空间的概率分布，假设 K_1 中选取的密钥和 K_2 中选取的密钥是相互独立的。因此，对任意的密钥 $k = (k_1, k_2) \in K_1 \times K_2$，有 $p(k_1, k_2) = p(k_1)p(k_2)$。

如果 $S_1 \times S_2 = S_2 \times S_1$，则称密码体制 S_1 与 S_2 是可交换的。注意，并不是所有的密码体制都是可交换的。

显然，密码体制的乘积运算满足结合律，即对任意具有相同明文和密文空间的密码体制 S_1、S_2、S_3，都有

$$(S_1 \times S_2) \times S_3 = S_1 \times (S_2 \times S_3)$$

设 S 是一个明文空间和密文空间相同的密码体制，定义：

$$S^n \overset{\text{def}}{=\!=} S_1 \times S_2 \times \cdots \times S_n$$

称为迭代密码体制。如果 $S^2 = S$，则称 S 是等幂的密码体制。

思考 3.4：在第 2 章中介绍的古典密码体制中，有哪些是等幂的密码体制？

置换密码、移位密码、仿射密码、Vigenere 密码、Hill 密码、Verman 密码等。 □

如果 S 是一个等幂的密码体制，则没有必要使用迭代体制 S^2，因为 S^2 使用更多的密钥但其安全强度却还与 S 一样。

如果 S 不是一个等幂的密码体制，则迭代密码体制 $S^n (n>1)$ 的安全强度会比 S 高。这种通过对一个密码体制进行迭代来提高密码体制安全强度的思想被广泛应用于分组密码体制的设计中（如 DES，参见 5.1 节）。因此，乘积密码的实现在现代密码体制的设计中具有重要的意义。

思考 3.5：如果 S_1 和 S_2 都是等幂的，且是可交换的，则 $S_1 \times S_2$ 也是等幂的。

$$\begin{aligned}(S_1 \times S_2)^2 &= (S_1 \times S_2) \times (S_1 \times S_2) = (S_1 \times S_2) \times (S_2 \times S_1) \\ &= S_1 \times (S_2 \times S_2) \times S_1 = S_1 \times S_2 \times S_1 = S_1 \times S_1 \times S_2 = S_1 \times S_2\end{aligned}$$

小结

本章首先利用信息论知识介绍了保密系统的数学模型和完善保密性。然后证明了一次一密的完善保密性，给出了完善保密性的充要条件。最后，讨论了通过密码系统的复用或者多个密码系统的联合来设计密码系统的思想，即乘积密码体制。

本章的重点在于：各个信息论基本概念间的关系，完善保密性，乘积密码体制。

扩展阅读建议

精读 Shannon 的两篇经典论文：

[1] SHANNON C E. A mathematical theory of communication [J]. Bell Labs Technical Journal,

1948，27(4)：379-423.
[2] SHANNON C E. Communication theory of secrecy systems [J]. Bell System Technical Journal，1949，28(4)：656-715.

习题

(1) 解释完善保密性。
(2) 解释英文的冗余度和唯一解距离。
(3) 解释乘积密码体制。
(4) 翻译两篇 Shannon 的论文中的一篇，并作 PPT 演讲。
(5) 撰写一篇报告，陈述对 Shannon 论文的读后感。

第4章　序列密码

本章要点：

- 序列密码的基本原理。
- 密钥流生成器，尤其是线性反馈移位寄存器。
- 案例学习：A5 算法。
- 案例学习：生成伪随机序列的软件方法 RC4 算法。

本章的学习次序视情况可以调整为，先学习两个案例（A5 算法和 RC4 算法），再学习基本原理。

序列密码（Stream Cipher）又称为流密码，是对称加密体制的一种。流密码对明文消息加密时，每次加密的单元要短一些，例如，序列密码每次可以加密 1 bit，分组加密的加密单元（如 DES）为 64 bit。这一点与分组加密每次加密一个“分组”区分开来。除此以外，分组密码中所有的加密单元（明文分组）都是用完全相同的加密函数和密钥来加密的，而流密码中所有的加密单元（如 1 bit）则是用相同的加密函数和不同的密钥来加密的。

相对分组密码而言，流密码的特点是：①在硬件实现上，速度一般比分组密码快，且不需要很复杂的硬件电路；②在某些情况下（如电信上的应用），当缓冲不足或必须对收到的字符进行逐一处理时，流密码显得更加必要和恰当；③流密码有较理想的数学分析工具，如频谱理论和技术、代数方法等。

序列密码由于具有坚实的数学基础和丰富的理论成果，因而广泛应用于军事、外交等国家重要部门的保密通信。Vernam 密码为流密码奠定了基础，“一次一密”的完善保密性证明导致了流密码的兴起。

4.1　序列密码的基本原理

在序列密码中，明文按一定长度分组后被表示成一个序列，称为明文流，序列中的一项称为“明文字”。加密时，先由主密钥产生一个密钥流序列，该序列的每一项和明文字具有相同的长度，称为一个“密钥字”。然后依次把明文流和密钥流中的对应项输入加密函数，产生相应的“密文字”，由密文字构成密文流输出。令

明文流为　　$M=m_1m_2\cdots m_i\cdots$

密钥流为　　$K=k_1k_2\cdots k_i\cdots$

加密算法为　　$C=c_1c_2\cdots c_i\cdots=E_{k_1}(m_1)E_{k_2}(m_2)\cdots E_{k_i}(m_i)\cdots$

解密算法为　　$M=m_1m_2\cdots m_i\cdots=D_{k_1}(c_1)D_{k_2}(c_2)\cdots D_{k_i}(c_i)\cdots$

序列密码进行保密通信的原理如图 4.1 所示。

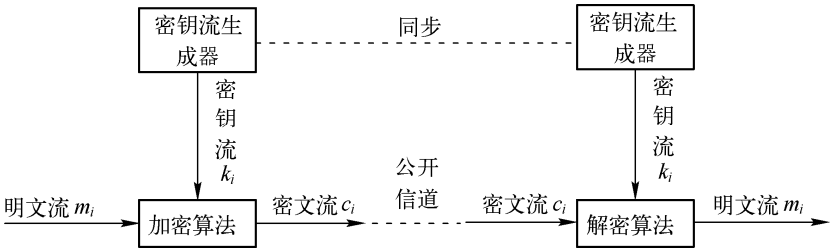


图 4.1　序列密码进行保密通信的原理图

4.1.1　序列密码的核心问题

第 3 章给出了“一次一密”具有无条件安全性的证明，从安全性来说它是理想的，但是，从实现效率来说，“一次一密”的一个明显的缺点就是它要求密钥与明文具有相同的长度。这增加了密钥分配、密钥生成和密钥管理的难度，这一缺陷极大地限制了它在实际中的应用。

使用完全随机的密钥流序列代价太大，一种自然的想法是使用伪随机（Pseudorandom）序列作为密钥流序列。从直观上讲，伪随机性（Pseudorandomness）意味着不是真正随机，但是很难将其和真正随机区分开来。

伪随机序列由伪随机生成器[㊀]（Pseudo-Random Generator，PRG）生成，PRG 以一定长度的比特串（称为种子密钥或 seed）作为输入，可输出任意长度的伪随机序列（也称为密钥流）。这样，若保密通信双方使用相同的 PRG，则只要传送种子即可。因此，基于 PRG 的序列密码克服了“一次一密”的密钥产生和传送困难的缺点，从而较为实用。虽然不具有无条件安全性，但增加了实用性，只要算法设计合理，其安全性可以满足实际应用的需要。

很自然地可以想到，序列密码的设计重点是**密钥流生成器**，其核心问题是如何衡量伪随机性和有效地生成伪随机序列。

4.1.2　序列密码的一般模型

序列密码的一般模型的关键是密钥流生成器。随机序列的产生由第 i 个时刻密钥流生成器的内部状态 σ_i 和种子密钥 k 确定，即

$$z_i = f(\sigma_i, k), i = 0, 1, \cdots$$

其中，f 是密钥流产生函数。

可见，流密码中的加密元件是有记忆的（这一点构成了流密码和分组密码的本质区别）。在分组密码（第 5 章介绍）中，加密密钥是时不变的，总是 k。而对流密码来说，加密密钥是时变的。真正加密的是 z_i，z_i 由函数 f、种子密钥 k 以及 i 时刻的状态 σ_i 确定。

分组密码和流密码的加密元件的区别如图 4.2 所示。

㊀　在加密场景中也称为密钥流生成器。

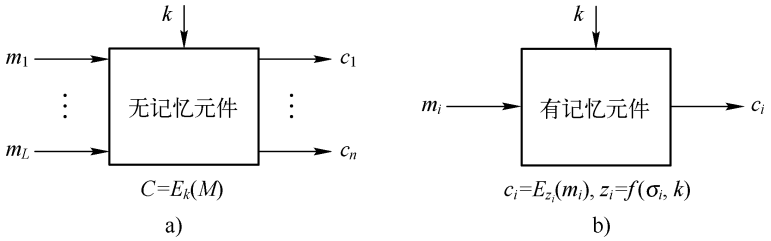


图 4.2　分组密码和流密码的本质区别（加密元件是否有记忆性）

a）分组密码　b）流密码

根据密码流生成器的内部状态 σ_i 是否与之前的密文有关，流密码可进一步分为两种类型：同步流密码（Synchronous Stream Cipher）和自同步流密码（Self-Synchronous Stream Cipher，又叫异步流密码）。

同步流密码系统如图 4.3 所示，内部状态与密文无关，所以密钥流与明文字符无关，从而 i 时刻的密文只取决于 i 时刻的明文，与 i 时刻前的明文无关。设 F 是流密码的状态转移函数，则有

$$\sigma_{i+1}=F(\sigma_i,k)$$
$$z_i=f(\sigma_i,k)$$

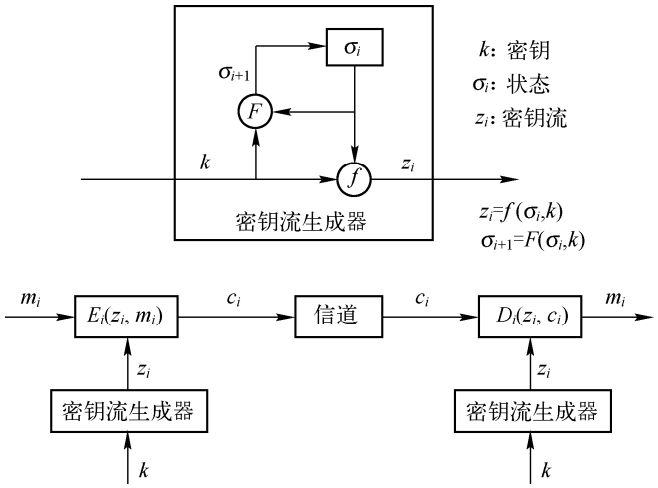


图 4.3　同步流密码系统

例 4.1　二元加法流密码是一种常见的同步流密码，即明文字符、密钥字，以及密文字符均为二元字符，且输出函数 E 为异或函数。二元加法同步流密码模型（Binary Additive Synchronous Stream Cipher），简称为二元流密码，是目前最为常见的流密码体制，也是一种常见的同步流密码（见图 4.4）。

$$c_i=m_i\oplus z_i$$

其中，m_i 为明文比特；z_i 为伪随机序列（密钥流）；c_i 为密文比特。取种子密钥为 k。

图 4.4 与第 2 章的图 2.1 相比，多出一个信道，这是因为安全需求（密码设计的动机）是通信的保密性，而不仅仅是保密性。“一次一密”密码是二元流密码的原型。事实上，如果 $z_i=k_i$（即密钥作为密钥流），则二元流密码退化为“一次一密”。

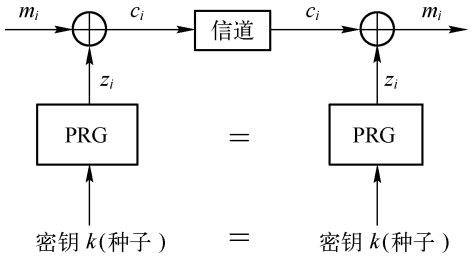


图 4.4　二元加法同步流密码系统（二元流密码）

同步流密码的特点如下：

（1）同步要求。在同步密码中，消息的发送者和接收者必须同步才能做到正确的加密和解密，即双方使用相同的密钥，并通过它对同一状态进行操作。一旦由于密文字符在传递过程中被插入或删除而破坏了这种同步性，那么解密工作将失败。这时只有借助其他方式重建同步，解密才能继续进行。

重置同步的技术包括：重新初始化，在密文的规则间隔中设置特殊符号，或如果明文包含足够的冗余度，就可以尝试密钥流的所有可能偏移。

（2）无错误传播。密文字符在传输过程中被修改（但未被删除和插入）并不影响其他密文字符的解密。

（3）主动攻击。作为性质（1）的结果，一个主动攻击者对密文字符进行的插入、删除或重复都会立即破坏系统的同步性，从而可能被解密器检测出来。作为性质（2）的结果，主动攻击者可能有选择地对密文字符进行改动，并准确地知道这些改动对明文的影响。因此，必须采用附加技术为数据提供数据源认证，并保证数据的完整性。

自同步流密码系统如图 4.5 所示，内部状态 σ_i 与密文有关，因此密钥流与明文字符有关，使得 i 时刻的密文不仅仅取决于 i 时刻的明文，而且与 i 时刻之前的 ℓ 个明文字符有关。设 F 是自同步流密码的状态转移函数，则有

$$\sigma_{i+1}=F(\sigma_i,c_i,c_{i-1},\cdots,c_{i-l},k)$$
$$z_i=f(\sigma_i,k)$$

或者

$$\sigma_{i+1}=F(\sigma_i,m_i,m_{i-1},\cdots,m_{i-l},k)$$
$$z_i=f(\sigma_i,k)$$

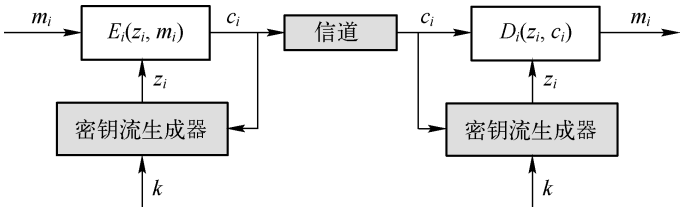


图 4.5　自同步流密码系统

例 4.2　这里构造一个特殊的例子用来说明自同步的特点。

假设密钥流就是上一个密文。

加密 m_1、m_2、m_3，密文为 c_1、c_2、c_3。其中

$$m_1 \oplus k = c_1$$
$$m_2 \oplus c_1 = c_2$$
$$m_3 \oplus c_2 = c_3$$

如果 c_1 被修改了，解密时，m_1 无法解出，m_2 无法解出，但是 m_3 仍然可以解密出来（通过 $c_3 \oplus c_2 = m_3$）。即 c_1 的删除影响两个明文，不影响第三个（及其以后）的明文。这便是一种自同步的特点。

思考 4.1：能否举一个当前加密密钥和前面两个密文有关的例子，从而理解错误扩散？ □

自同步流密码的特点如下：

（1）自同步。由于对当前密文字的解密仅仅依赖于固定个数的以前的密文字，因此，当密文字被插入或者删除时，密码的自同步性就会体现出来。这种密码在同步性遭到破坏时，可以自动地重建正确的解密，而且仅有固定数量的明文字不可恢复。

（2）有限的错误传播。假设一个自同步流密码系统的状态依赖于 t 个以前的密文字。在传输过程中，当一个单独的密文字被改动（或被插入、删除）时，至多有 t 个随后的密文字解密出错，然后恢复正确解密。

（3）主动攻击。从性质（2）看出，主动攻击对密文字的任何改动都会引发一些密文字的解密出错。因此，与同步流密码相比，自同步流密码具有更高的被（解密器）检测出来的可能性。作为性质（1）的结果，这种密码在检测主动攻击者发起的对“密文字”的插入、删除、重复等攻击时，就会更加困难，必须采用一些附加的技术来实现消息源鉴别和消息完整性。

（4）明文统计扩散。每个明文字都会影响其后的整个密文，即明文的统计学特征被扩散到了密文中。因此，自同步流密码在抵抗利用明文冗余度而发起的攻击方面要强于同步流密码。

思考 4.2：比较同步流密码和自同步流密码的优缺点。

同步流密码中，密（明）文字符是独立的，一个错误传输只会影响一个字符，不会影响后面的字符。它的优点是容易检测插入、删除、重复等主动攻击，且没有差错传播。但是，一旦接收端及发送端的种子密钥和内部状态不同步，解密就会失败，两者必须借助外界手段才能重新建立同步。

自同步流密码中，密（明）文字符参与了密钥流的生成，一个错误传输将影响后面 ℓ 个字符。与同步密钥流相比，其优点是即使接收端和发送端不同步，只要接收端能连续接收到 ℓ 个密文字符，就能重新建立同步。因此，自同步流密码具有有限的差错传播，且能把明文的每个字符都扩散在密文的多个字符中，从而强化了抗统计分析的能力。 □

4.2 密钥流生成器

4.2.1 密钥流生成器的架构

上节讨论了序列密码对密钥流的安全性要求。通常安全性要求越高，设计越复杂，因此在设计密钥流生成器时，既要考虑安全性，也要考虑实用性：①密钥 K 容易分配、保管、

更换；②易于实现，快速。

首先复习一下有限状态自动机理论，该理论可以指导密钥流生成器的设计。

有限状态自动机是具有离散输入和输出（输入集和输出集）的一种数学模型，它由以下 3 个部分组成：

（1）有限状态集：

$$S=\{s_i \mid i=1,2,\cdots,l\}$$

（2）有限输入字符集：

$$A=\{A_j \mid j=1,2,\cdots,m\}$$

有限输出字符集：

$$B=\{B_k \mid k=1,2,\cdots,n\}$$

（3）转移函数：

$$B_k=f_1(s_i,A_j),s_h=f_2(s_i,A_j)$$

即在状态为 s_i，输入为 A_j 时，状态转换为 s_h，并输出一个字符 B_k。

例 4.3 $S=\{s_1,s_2,s_3\}$，$A=\{A_1,A_2,A_3\}$，$B=\{B_1,B_2,B_3\}$，转移函数由表 4.1 给出。

表 4.1 转移函数 f_1 和 f_2

f_1	A_1	A_2	A_3
s_1	B_1	B_3	B_2
s_2	B_2	B_1	B_3
s_3	B_3	B_2	B_1
f_2	A_1	A_2	A_3
s_1	s_2	s_1	s_3
s_2	s_3	s_2	s_1
s_3	s_1	s_3	s_2

有限状态自动机可用有向图表示，称为状态转移图。顶点表示状态，有向弧表示输入输出字符，如图 4.6 所示。

若输入序列是：$A_1A_2A_1A_1A_1$，初始状态为 s_1，则输出序列为 $B_1B_1B_2B_3B_1$。

以同步流密码的密钥流产生器为例。将其视为一个参数为 k 的有限状态自动机进行分析，它由一个输出符号集 Z、一个状态集 Σ、两个状态转移函数 ϕ 和 ψ 以及一个初始状态 σ_0 组成，如图 4.7 所示。

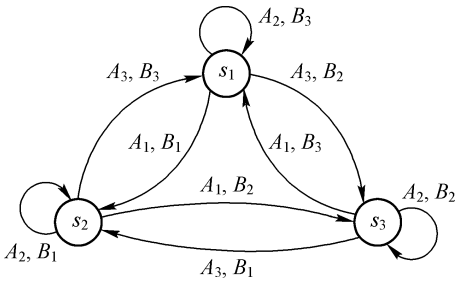


图 4.6 有限状态自动机的状态转移图

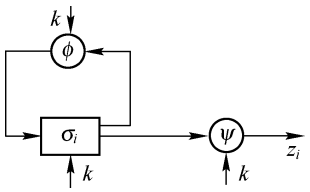


图 4.7 密钥流生成器的有限状态自动机

其中状态转移函数为 $\phi:\sigma_i\to\sigma_{i+1}$，$\psi:\sigma_i\to z_i$。于是，密钥流生成器设计的重点就是找出适当的状态转移函数 ϕ 和 ψ，使得输出序列 z 满足安全条件和实用条件。

由于非线性 ϕ 的有限状态自动机理论很不完善，所以相应的密钥流产生器的分析会受到限制。相反地，当采用线性的 ϕ 和非线性的 ψ 时，能够进行深入的分析，并可以得到好的生成器。通常线性部分称为驱动子系统，非线性部分称为非线性组合子系统。驱动子系统控制生成器的状态转移，并为非线性组合部分提供统计性能良好的序列；非线性组合部分利用这些序列组合出满足要求的密钥流序列。

常见的产生密钥流的方法有线性反馈移位寄存器、非线性反馈移位寄存器、有限自动机等方法，以及近年来提出的混沌密码技术。目前最流行和实用的密钥流产生器如图 4.8 所示，其驱动部分是一个或者多个线性反馈移位寄存器（Linear Feedback Shift Register，LFSR）。线性反馈移位寄存器结构简单，非常适合硬件实现，运行速度快，同时可以产生大周期序列和良好统计性质的序列。下一节将详细介绍。

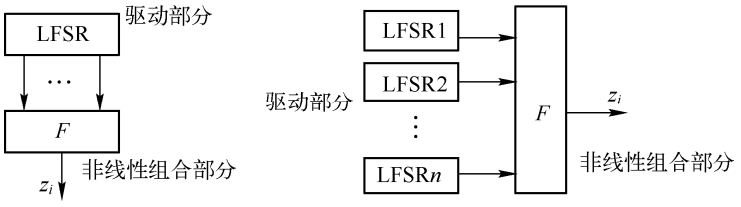


图 4.8　常见的两种密钥流生成器

4.2.2　线性反馈移位寄存器

线性反馈移位寄存器（LFSR）是序列密码中密钥流生成器的一个重要组成部分，简言之，一个线性反馈移位寄存器由两个部分组成：移位寄存器（Shift Register）和反馈函数（Feedback Function）。移位寄存器是位序列，具有 n 位长的移位寄存器称为 n 位移位寄存器。每次输出一位，然后寄存器中的所有位都右移一位。新的最左端的位根据寄存器中的其他位计算得到，寄存器输出的一位通常是最低有效位。通常反馈函数是寄存器中某些位的简单异或，这些位叫抽头序列（Tap Sequence），有时也叫 Fibonacci 配置。

下面更一般性地介绍线性反馈移位寄存器的基本性质和基本结论。

1. 反馈移位寄存器

反馈移位寄存器（FSR）是由时钟控制的若干串联的寄存器组成的，寄存器的个数称为 LSFR 的级数。在时钟控制下，寄存器中存储的信息依次由上一级向下一级传递，而寄存器中全部信息经过某种运算后，反馈回来作为第一级寄存器的输入。

如图 4.9 所示为一个 n 级 FSR 模型。

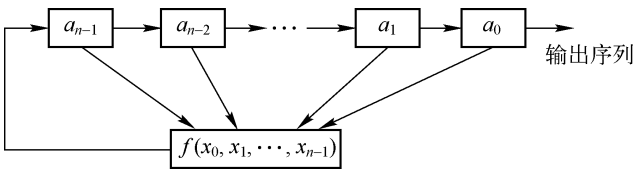


图 4.9　FSR 模型

从左到右，寄存器依次称为第 1 级，第 2 级，…，第 n 级，其中存放的信息分别为 $a_{n-1},a_{n-2},\cdots,a_0$，记为$(a_0,a_1,\cdots,a_{n-1})$，称为移位寄存器的状态。开始工作前具有的状态称为初始状态。当第一个时钟脉冲到达时，上一级寄存器中的信息右移至下一级寄存器中，而最后一个寄存器中的信息 a_0 移出寄存器作为输出，同时，寄存器中全部信息 $a_{n-1},a_{n-2},\cdots,a_0$ 作为 n 个自变量，经过函数 f 运算后得到新信息，记为 a_n，反馈进入第 1 级寄存器。从而寄存器状态变为$(a_1,a_2,\cdots,a_n)$。依此类推，随着时钟脉冲的变化，FSR 将输出序列 $a=(a_0,a_1,\cdots)$，序列 a 被称为移位寄存器序列，它满足以下递推公式：

$$a_{n+k}=f(a_{n+k-1},a_{n+k-2},\cdots,a_k),k=0,1,\cdots$$

f 称为 FSR 的反馈函数。如果反馈函数 f 是一个 n 元线性函数，即

$$f(x_1,x_2,\cdots,x_n)=c_1x_1+c_2x_2+\cdots+c_nx_n,c_i\in\{0,1\}$$

则序列 a 被称为线性反馈移位寄存器（Linear Feedback Shift Register，LFSR）序列，否则序列 a 被称为非线性反馈移位寄存器（Non-Linear Feedback Shift Register，NLFSR）序列。

2. 特征多项式

显然，反馈函数确定了生成的序列，于是，研究的重点转移到反馈函数的性质。

LFSR 序列 $a=(a_0,a_1,a_2,\cdots)$满足

$$a_{n+k}=c_1a_{n+k-1}+\cdots+c_{n-1}a_{k+1}+c_na_k,k=0,1,\cdots$$

由于是二元域，可以移项写为

$$a_{n+k}+c_1a_{n+k-1}+\cdots+c_{n-1}a_{k+1}+c_na_k=0,k=0,1,\cdots$$

为了方便，可用一元 n 次多项式

$$f(x)=1+c_1x_1+\cdots+c_nx_n$$

来表示反馈函数，$f(x)$称为 LFSR 序列 a 的特征多项式（Characteristic Polynomial）。

顾名思义，当给定特征多项式和初始态后，LFSR 的整个输出序列就完全确定了。固定特征多项式时，不同的初始态可以确定不同的序列。由于有 2^n 种初始态，故最多有 2^n 个序列。

为了直观地表示反馈函数，可以使用 LSFR 电路图，如图 4.10 所示。其中开关线路 c_i 称为相应寄存器的抽头，当 $c_i=1$ 时，开关 c_i 合上；否则，开关 c_i 断开。反馈信息就是对各抽头信息进行运算得到的。

如果 $c_n=0$，则第 n 级寄存器（即 a_0）对输出没有影响，这等价于降低了寄存器的级数。为了保持是“n 级”LSFR，总是要求 $c_n=1$。

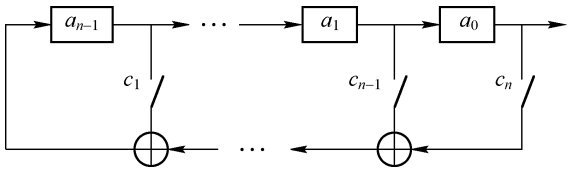


图 4.10　LFSR 的电路图表示法

例 4.4　3 级 LFSR 的特征多项式为 $f(x)=1+x^2+x^3$，初始态为（0,0,1）。求输出序列及其周期。

画出 3 级 LFSR 示意图，如图 4.11 所示。

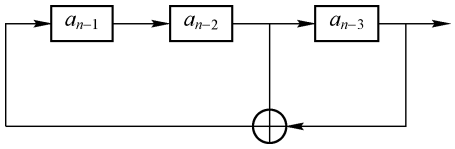


图 4.11　3 级 LFSR 示意图

输出序列 $a=(a_0,a_1,a_2,\cdots)$ 满足递推关系式 $a_n=a_{n-2}+a_{n-3}(n=3,4,\cdots)$，根据递推关系，求出输出序列为 0010111，周期为 7。

思考 4.3：

3 级 LFSR 的特征多项式为 $f(x)=1+x+x^3$，初始态为（0,0,1）。求输出序列及其周期。

3 级 LFSR 的特征多项式为 $f(x)=1+x+x^2+x^3$，初始态为（0,0,1）。求输出序列及其周期。

2 级 LFSR 的特征多项式为 $f(x)=1+x^2$，初始态为（1,1）。求输出序列及其周期。

2 级 LFSR 的特征多项式为 $f(x)=1+x+x^2$，初始态为（1,1）。求输出序列及其周期。 □

思考 4.4：为什么需要假定 $c_i(1\leqslant i\leqslant n)$ 中至少有一个不为 0?

如果 c_i 全为 0，则无抽头，没有反馈，必然在 n 个脉冲后状态变为 0，且这个状态必将一直持续下去。 □

思考 4.5：为什么 LFSR 序列会有周期?

n 级 LFSR 的状态数最多为 2^n，因此，从初始态到第 2^n+1 个状态中，必有两个状态相同。由于 LFSR 由当前状态和特征多项式决定，故两个相同的状态必导致它们后续的状态相同。故生成周期序列。周期为两个相同状态间的状态数加 1。 □

思考 4.6：n 级 LFSR 序列的最大周期是多少?

n 级 LFSR 序列的总状态数为 2^n。如果初始态为全 0，则其状态恒为 0。如果初始态非 0，则其后继状态也不会为 0（思考一下为什么）。于是 0 状态不会计算在内，故最大周期为 2^n-1。 □

周期为 2^n-1 的 n 级 LFSR 序列称为最大长度（Maximal Length）序列，简称 m 序列。下面这个定理说明如何找到 m 序列。

定理 4.1　序列 a 是周期为 2^n-1 的 m 序列的充要条件是其特征多项式 $f(x)$ 为 n 阶本原多项式。

定理的证明留作练习。

4.2.3　非线性序列生成器*

密钥流生成器分为驱动和非线性组合两个部分，前面介绍了 LFSR 部分，该方面的理论非常成熟，人们已经得到了具有良好伪随机特性的序列，如 m 序列等。因此，非线性组合的设计就成为密钥流生成器设计的关键问题。

1. 非线性准则

一般地，非线性组合部分可以由布尔函数表示，于是对非线性组合部分的研究可以归结为对布尔函数的研究。随着对非线性布尔函数研究的深入，人们提出了许多非线性设计的准

则：代数次数、非线性度、相关免疫性、退化性、雪崩准则、扩散准则等。

（1）代数次数要尽可能大。

n 元布尔函数 $f(x)$ 是从 Z_2^n 到 Z_2 的一个映射，它可以由如下多项式表示：

$$f(x_1,x_2,\cdots,x_n)=a_0+a_1x_1+a_2x_2+\cdots+a_nx_n+a_{1,2}x_1x_2+\cdots+a_{n-1}x_{n-1}x_n+a_{1,2,3}x_1x_2x_3+\cdots+a_{1,2,\cdots,n}x_1x_2\cdots x_n$$

其中所有的系数都是 0 或者 1。该多项式也称为布尔函数 $f(x)$ 的代数标准型。每项中变量的个数称为该项的次数。布尔函数 $f(x)$ 的代数次数定义为 $f(x)$ 的代数标准型中具有非零系数的乘积项的最大次数。

当 $f(x)$ 的代数次数为 1 时，$f(x)$ 为线性布尔函数；当 $f(x)$ 的代数次数大于 1 时，$f(x)$ 称为非线性布尔函数。显然，非线性组合部分的布尔函数应该具有尽可能大的代数次数。

（2）非线性度要尽可能大。

设 L 是 Z_2^n 上所有线性函数的集合，即 $L=\{u\cdot x+v \mid u\in Z_2^n, v\in Z_2\}$，则布尔函数 $f(x)$ 的非线性度定义为

$$N_f=\min_{l(x)\in L} d_H(f(x),l(x))$$

其中，$d_H(f(x),l(x))$ 是函数 $f(x)$ 和 $l(x)$ 的海明距离；$u\cdot x$ 是点积运算。

非线性度用海明距离度量布尔函数和线性函数的相似程度，它是刻画密码系统抵抗线性攻击的一个重要指标。

（3）避免退化。

设 $f(x)$ 是一个 n 元布尔函数，如果存在 Z_2 上一个 $k\times n(k<n)$ 的矩阵 $\boldsymbol{D}$，使得 $f(x)=g(\boldsymbol{D}x)=g(y)$，则称 $f(x)$ 是退化的。

布尔函数 $f(x)$ 经过自变量的线性变换后简化为 k 元布尔函数 $g(x)$，降低了安全性。

（4）具备 m 阶相关免疫性。

设 $f(x_1,x_2,\cdots,x_n)$ 是 n 个彼此独立且对称的二元随机变量的布尔函数，称其为 m 阶相关免疫，当且仅当任意 $f=f(x_1,x_2,\cdots,x_n)$ 与 $x_1,x_2,\cdots,x_n$ 中的任意 m 个随机变量 $(x_{i_1},x_{i_2},\cdots,x_{i_m})$ 统计无关，即对任意 $(a_1,a_2,\cdots,a_m)\in Z_2^m$ 和 $a\in Z_2$，$f(x_1,x_2,\cdots,x_n)$ 时，满足

$$\Pr(f=a,x_{i_1}=a,x_{i_2}=a_2,\cdots,x_{i_m}=a_m)=\frac{1}{2^m}\Pr(f=a)$$

相关免疫性是为防止攻击者对密码系统进行相关攻击而提出的指标。可用 Walsh 变换刻画相关免疫性。

（5）满足严格雪崩准则。

如果对于任意 $w_{\mathrm{H}}(e)=1$ 的 $e=(e_1,e_2,\cdots,e_n)\in Z_2^n$，都有 $f(x)+f(x+e)$ 是平衡函数，则称 n 元布尔函数 $f(x)$ 满足严格雪崩准则。这里 $w_{\mathrm{H}}(\cdot)$ 是海明重量。

（6）满足 m 次扩散准则。

设 $1\leqslant m\leqslant n-2$，如果任意 $1\leqslant w_{\mathrm{H}}(e)\leqslant m$ 的 $e=(e_1,e_2,\cdots,e_n)\in Z_2^n$，都有 $f(x)+f(x+e)$ 是平衡函数，则称 n 元布尔函数 $f(x)$ 满足 m 次扩散准则。

2. 非线性组合

非线性组合主要有 3 种方式：非线性组合方式、非线性滤波生成器和钟控方式。

非线性组合方式将 n 个 LFSR 的输出作为一个非线性函数的输入，最后输出结果。如图 4.12a 所示。

非线性滤波生成器（Filter Generator，FG）又称为前馈生成器，将一个 LFSR 的各位通

过一个非线性函数的组合输出。如图 4.12b 所示。

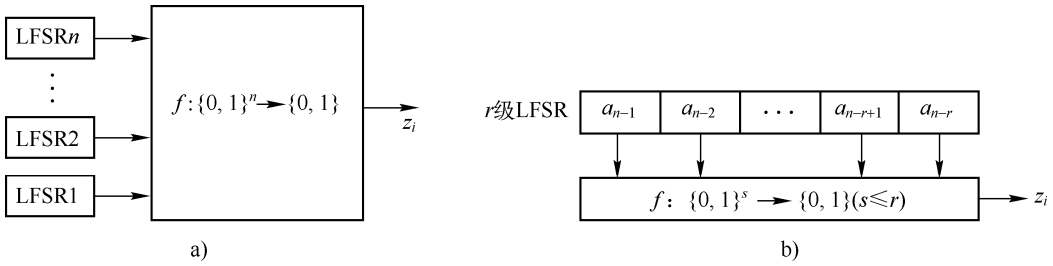


图 4.12　非线性组合方式和非线性滤波生成器

a）非线性组合生成器　b）非线性滤波生成器

钟控生成器（Clock Control Generator，CCG）的基本原理是一些 LFSR 在另外一些 LFSR 的控制下以不规则的时钟输出。

例如，停-走式钟控生成器（Stop-and-Go Generator）的规则是：设控制 LFSR 和种子 LFSR 在正常时钟下的输出序列分别为 $a_0,a_1,\cdots$和 $b_0,b_1,\cdots$，停-走式钟控生成器的生成序列为 $z_0,z_1,\cdots$，则 $z_i=b_{\sigma(i)}$，其中 $\sigma(i)=\sum_{k<i}a_k$，$\sigma(0)=0$。

4.2.4　案例学习：A5 算法

A5 算法是欧洲移动通信系统 GSM 采用的流密码算法，用于加密从手机到基站间的语音通信。GSM 会话的帧长为 228 bit，A5 算法的密钥长 64 bit，每次产生长度为 228 bit 的会话密钥。

A5 有两个版本：A5/1 和 A5/2，前者安全性更高，根据相关法规用于欧洲内部，后者用于其他地区。

A5 算法由 3 个长度分别为 19、22、23 的 LFSR 组合成钟控密钥流生成器。LFSR 为常用硬件（寄存器），处理速度快，价格便宜。

LFSR1 的抽头是 13、16、17、18（单元从 0 开始编号），特征多项式为 $f(x)=1+x^{14}+x^{17}+x^{18}+x^{19}$；

LFSR2 的抽头为 12、16、20、21，特征多项式为 $f(x)=1+x^{13}+x^{17}+x^{21}+x^{22}$；

LFSR3 的抽头为 17、18、21、22，特征多项式为 $f(x)=1+x^{18}+x^{19}+x^{22}+x^{23}$。

A5/1 的 LFSR 时钟信号由钟控函数 $g(x,y,z)=xy+xz+yz$ 的值决定，其中 x、y、z 分别为 LFSR1、LFSR2、LFSR3 的中间位置单元（即第 9、11、11 个单元）的值。

当 x 和 $g(x,y,z)$ 的值相等时，LFSR1 移位，否则重复输出前一位。LFSR2 和 LFSR3 的移位过程与 LFSR1 类似。算法如图 4.13 所示。

A5 算法的工作过程如下：

（1）将长度为 64 bit 的密钥输入 LFSR，因为 3 个 LFSR 的总长度为 64 bit。

（2）将 22 bit 的帧数与 LFSR 的反馈值模 2 加，再输入 LFSR。

（3）LFSR 开始停-走钟控。

（4）舍去产生的 100 bit 输出。

（5）产生 114 bit 作为密钥流。

（6）舍去产生的 100 bit 输出。

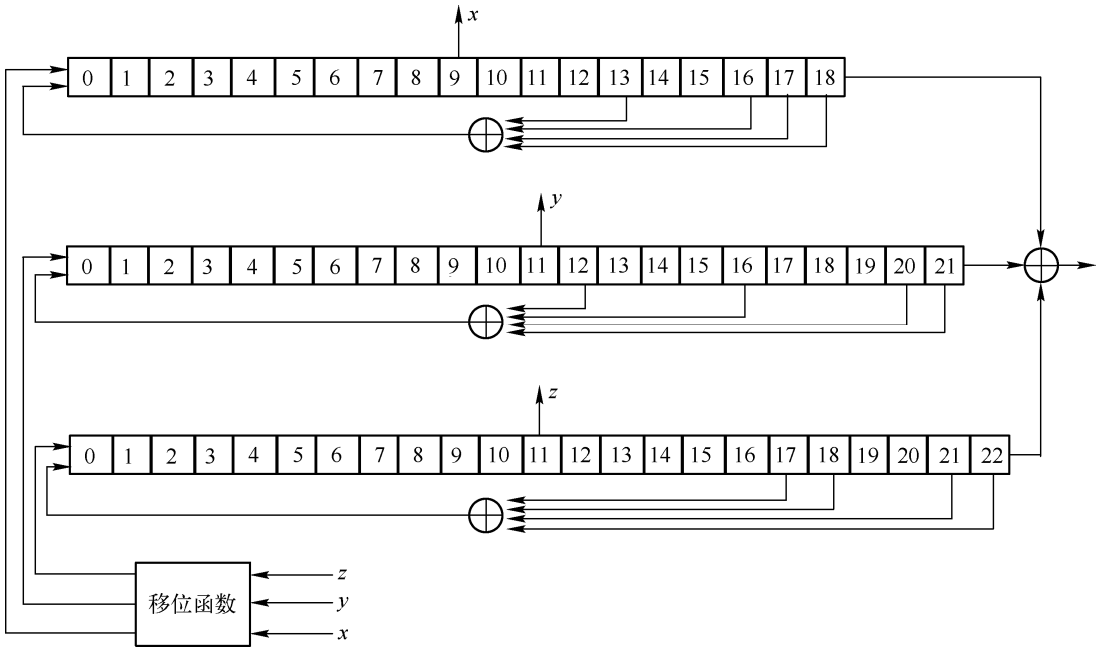


图 4.13　A5 算法

（7）产生 114 bit 作为密钥流。

故共产生 228 bit 密钥流。

A5 算法具有良好的统计特性，同时效率较高，但后来研究发现，A5 中的钟控机制存在一些细微的设计缺陷，导致密钥流序列周期较短。同时，不同的种子密钥可能产生相同的密钥流，导致密钥冲突。（因此，3G 移动通信中建议的数据加密算法 f8，是基于分组密码 KASUMI 算法构造随机序列的方法。）

4.3　伪随机序列生成器的其他方法

前面介绍了通过 LFSR 构造伪随机序列生成器。其实，还有如下其他几种构造的方法：

① 通过分组密码的序列化实现，即通过分组密码加密生成随机序列，一般通过分组密码的加密模式进行（详见第 5.5 节分组密码的工作模式）。

② 基于软件实现的快速加密算法，如 RC4、SEAL、SCREAM 等算法，这些算法的设计各有特殊的巧妙之处。

③ 基于计算复杂性的方法，通过模运算构造，典型的方法是 Blum-Blum-Shub 生成器和 RSA 生成器。这类生成器的优点是提供可证明安全性，即将安全性和数学难题相关联，安全性的破坏必然导致公认数学难题的求解。但是，实现时计算量太大，常用于随机数生成器（详见 11.2.2 节，密码学上安全的伪随机比特生成器）。

④ 基于混沌理论的方法。

4.3.1　基于软件实现的方法（案例学习：RC4 算法）

基于软件实现的方法主要是 RC4 算法。RC4 算法由 R. Rivest 于 1987 年提出（RC4 表示“Rivest Cipher 4”），已被广泛应用于 Windows 等软件和安全套接字层（Secure Socket Layer，SSL）、无线局域网络安全协议（Wired Equivalent Privacy，WEP）等，是目前公开范

围内应用最广泛的序列密码。

RC4 算法以随机置换为基础，具有可变密钥长度，面向字节（Byte）操作。其整体实现是：先用不大于 256 B 的可变长密钥初始化一个 256 B 的状态数组变量 S，S 的元素记为 $S[0],S[1],\cdots,S[255]$。然后，对 S 中的字节进行适当置换，置换后的 S 始终包含 0~255 所有的 8 bit 数，每次置换后产生 1 B 密钥。

具体过程如下：

1. 初始化 *S*

首先，为 S 赋初始值 $S[i]=i(0\leqslant i\leqslant 255)$，并建立长度为 256 B 的临时数组变量 T，再重复用长度为 Keylen 字节的密钥 K 对 T 赋值，直到 T 的所有元素被赋值；然后用 T 对 S 进行初始置换，对每个 $S[i]$，由 $T[i]$将其置换为 S 中的另一个字节。代码如下：

```
for i=0 to 255
    S[i]=i
    T[i]=K[i mod Keylen]
j=0
for i=0 to 255
    j=(j+S[i]+T[i]) mod 256
    Swap(S[i],S[j]) /* 交换两个字节 */
```

2. 随机序列生成（密钥流生成）

初始化完成后，丢弃密钥 K，用 $S[0]\sim S[255]$产生密钥序列：根据当前 S 的值，将 $S[i]$与S 中的另一字节 $S[j]$置换，置换完成后，由 $S[i]$和 $S[j]$的值产生长度为 1 Byte 的密钥 K。当全部 256 B 完成置换后，从 $S[0]$开始继续重复，直到产生实际需要长度的密钥为止。代码如下：

```
i,j=0
While (true)
   i=(i+1) mod 256
   j=(j+S[i]) mod 256
   Swap(S[i],S[j])
   t=(S[i]+S[j]) mod 256
   K=S[t]
```

输出的字节先查找 $S[i]$和 $S[j]$的值，两个值相加后模 256，作为索引，查找 S(S[i] + [j]) 得到密钥流 K。计算过程的示意图如图 4.14 所示。

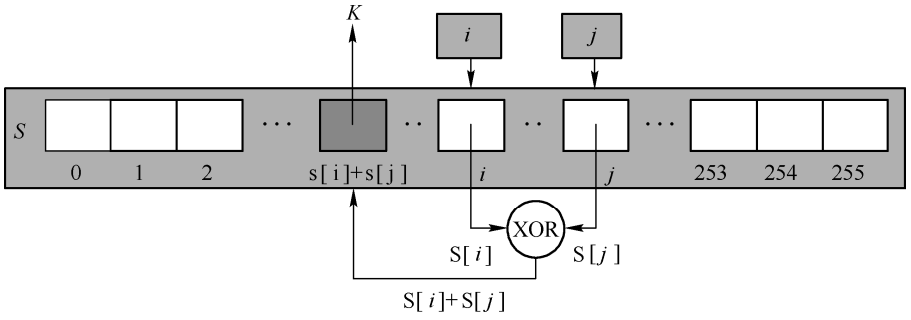


图 4.14　RC4 算法密钥流产生过程示意图

RC4 的优点是算法简单、高效、特别适合软件实现，加密速度比 DES（将在第 5 章介绍）快约 10 倍。

下面给出 RC4 算法的 C 语言源代码。

```
/*********************************************
 * RC4 算法实例程序
 * 作者:任伟
 *********************************************/

#include <stdio.h>
#include <string.h>
//全局变量声明
int S[256];                                    //状态数组
char * key;                                    //密钥数组
int RAND[256];                                 //随机密钥流

void KSA(char * key);                          //密钥调度算法
void PRGA();                                   //伪随机数造成算法

void Swap_State(int i,int j);                  //辅助函数声明

void Swap_State(int i,int j)                   //交换 S[i]和 S[j]
{
        int temp;
        temp = S[i];
        S[i] = S[j];
        S[j] = temp;
}

void KSA(char * key)
{

        int j;                                 //索引
        int keylength;
        int i;

        keylength = strlen(key);

        for (i=0;i<256;i++)                    //初始化状态数组
                S[i]=i;

        j=0;

        for (i=0;i<256;i++)
        {
                j=(j+S[i]+key[i%keylength])%256;
                Swap_State(i,j);

                //printf("S[%i]=%d\n",i,S[i]); //调试输出
        }
```

```
}

void PRGA()
{
        //初始化
        int i=0;
        int j=0;

        int number=0;                                    //已经生成的随机字节的个数

        //通过循环生成
    while (number<256)
        {
    i=(i+1)%256;
    j=(j+S[i])%256;

    Swap_State(i,j);

        RAND[number] = S[(S[i]+S[j])%256];

        //printf("RAND[%i]=%d\n",number,RAND[number]);  //调试输出
        number++;
       }
}

int main()                                               //测试 RC4 算法
{
       char * plaintext;
       unsigned int ciphertext[256];
       int i;
       int plaintext_length;                             //明文的长度


       plaintext = "Plaintext";                          //三个来自 Wikipedia 的测试数据
       //plaintext = "pedia";
       //plaintext = "Attack at dawn";
       //plaintext ="Network Security";

       key = "Key";
       //key = "Wiki";
       //key = "Secret";


       plaintext_length =strlen(plaintext);

       KSA(key);
       PRGA();

       printf("RC4 Experiment, June 10 2009, Wei Ren, School of Computer Science, CUG. \n\n");
       for(i=0; i<plaintext_length; i++)
```

```
        {
                ciphertext[i] = plaintext[i] ^ RAND[i];                //按位异或

                printf("plaintext[%i]=%c\t",i,plaintext[i]);
                printf("ciphertext[%i]=%x\n",i,ciphertext[i]);         //用十六进制数标记
        }

        return 0;
}

/**********************************************************************
测试数据来自 Wikipedia
http://en.wikipedia.org/wiki/RC4

RC4("Key", "Plaintext") == BBF316E8D940AF0AD3
RC4("Wiki", "pedia") == 1021BF0420
RC4("Secret", "Attack at dawn") == 45A01F645FC35B383552544B9BF5
**********************************************************************/
```

4.3.2 基于混沌的方法简介

Robert A. J. Matthews 在 1989 年首次将混沌理论用于密码学研究，并提出一种基于变形 Logistic 映射的混沌序列密码方案。从此，混沌密码学作为密码学的一个分支引起了广泛的关注。美国海军实验室研究人员 Pecora 和 Carroll 首次利用驱动-响应法实现了两个混沌系统的同步，这一突破性的研究成果为混沌理论在通信的应用开辟了道路。1997 年后，密码学研究界掀起了数字混沌密码学研究的高潮。

混沌系统具有高频谱、类随机特性、对结构参数及初始状态的极端敏感性等一系列性质，使得混沌成为密码学研究的一个重要领域。混沌作为一种非线性现象，有许多独特的且值得利用的性质，或许能够为密码学的发展提供新的思路。

小结

本章首先介绍了序列密码的一般模型，研究的核心问题是伪随机序列的要求。然后，介绍了密钥流生成器，包括密钥流生成器的架构、LFSR、NLFSR，并给出一个实例（A5 算法）。最后，介绍了其他伪随机序列生成器的方法，如基于软件实现的方法，实例为 RC4 算法。本章要点总结如下。

本章的重点是密钥流生成器的 LFSR、A5 算法、RC4 算法。难点是同步流密码系统和非线性序列生成器。

扩展阅读建议

ECRYPT 项目建立了流密码工程 eStream，并于 2008 年 9 月最终选择了 HC-128、Rabbit、Salsa20/12、SOSEMANUK 这 4 个适合软件实现的算法，以及 Grain v1、MICKEY 2.0 和 Trivium 这 3 个适合硬件实现的算法。相关算法查阅 http://www.ecrypt.eu.org/stream/。

本章的相关文献有：

[1] 李超，等．密码函数的安全性指标分析［M］．北京：科学出版社，2011.
[2] 陈智雄．伪随机序列的设计及其密码学应用［M］．厦门：厦门大学出版社，2011.
[3] 廖晓峰，等．混沌密码学原理及其应用［M］．北京：科学出版社，2009.
[4] 胡予濮，张玉清，肖国镇．对称密码学［M］．北京：机械工业出版社，2002.
[5] 李晖，李丽香，邵帅．对称密码学及其应用［M］．北京：北京邮电大学出版社，2009.

上机实验

（1）用 FPGA 实现 A5 算法。

（2）用 Python、C、Java 等编程语言实现 RC4 算法。

习题

（1）构造一个例子，说明自同步密码的自同步特性。

（2）画同步流密码系统图和自同步流密码系统图，并指出其区别。

（3）为什么自同步流密码可以做到自同步和有限的错误传播？举例说明。

（4）A5 算法中的钟控函数是什么，它是如何控制 LSFR 的？

（5）A5 算法中 LSFR 的特征函数分别是什么？

（6）RC4 算法中的 KSA 和 PRGA 分别表示什么，它们各做了哪些工作？

（7）4 级 LFSR 的特征多项式为 $f(x)=1+x^2+x^4$，初始态为（0,0,1,1）。求输出序列及其周期。

（8）4 级 LFSR 的特征多项式为 $f(x)=1+x^3+x^4$，初始态为（0,0,1,1）。求输出序列及其周期。

（9）4 级 LFSR 的特征多项式为 $f(x)=1+x+x^4$，初始态为（0,0,1,1）。求输出序列及其周期。

（10）求一个 4 级 LFSR 的特征多项式 $f(x)$，其初始态为（0,0,1,1），使得输出序列的周期为最大值 15。

（11）设计题：设计一个以 LSFR 为基础的算法，以自己的名字命名。算法中有两个 LSFR，特征分别是：$f(x)=1+x^4+x^5$，$f(x)=1+x^3$，钟控抽头为寄存器的中间比特，钟控函数是 $g(x,y)=x*y$，初始密钥是 11001 ‖ 110。输出前 30 bit，加密自己的名字。

（12）研究题：研究通过混沌产生流密码的方法，并加密一个多媒体数据。

（13）研究题：研究构造 m 序列的方法，撰写一份研究报告。